SYNGRESS®

CompTIA Linux+ Certification Study Guide

Exam XK0-003

Brian Barber

- Pass the exam the first time
- Filled with exercises, real-world examples, questions, and answers
- DVD-ROM contains two full-length practice exams to help you prepare for test day

CompTIA Linux+ Certification Study Guide

Exam XK0-003

This page intentionally left blank

CompTIA Linux+ Certification Study Guide

Exam XK0-003

Brian Barber

Technical Editor
Kevin Riggins

Contributing Authors
Chris Happel
Terrence V. Lillard
Graham Speake

ELSEVIER

AMSTERDAM • BOSTON • HEIDELBERG • LONDON
NEW YORK • OXFORD • PARIS • SAN DIEGO
SAN FRANCISCO • SINGAPORE • SYDNEY • TOKYO

Syngress is an imprint of Elsevier

SYNGRESS®

Syngress is an imprint of Elsevier
30 Corporate Drive, Suite 400, Burlington, MA 01803, USA
Linacre House, Jordan Hill, Oxford OX2 8DP, UK

CompTIA Linux+ Certification Study Guide: Exam XK0-003

Library of Congress Cataloging-in-Publication Data
Application submitted

British Library Cataloguing-in-Publication Data
A catalogue record for this book is available from the British Library.

ISBN: 978-1-59749-482-3

Printed in the United States of America
09 10 11 12 13 10 9 8 7 6 5 4 3 2 1

For information on rights, translations, and bulk sales, contact Matt Pedersen, Commercial Sales Director and Rights; email: *m.pedersen@elsevier.com*

For information on all Syngress publications visit our Web site at *www.syngress.com*

Contents

About the Authors

LEAD AUTHOR

Brian Barber (Linux+, MCSE, MCSA, MCP+I, MCNE, CNE, CNA-GW) works for the Canada Deposit Insurance Corporation (CDIC) as a project manager and architect for CDIC's IT service management program. He first started using Linux at home with Red Hat 5.1 and since then he has been a staunch advocate of open-source software, belonging to the Ottawa Canada Linux User Group (OCLUG) since 2001, and the Ottawa Python Authors Group. His primary areas of interest are operating systems, infrastructure design, multiplatform integration, directory services, and enterprise messaging. In the past, he has held the positions of Principal Consultant with Sierra Systems Group Inc., Senior Technical Coordinator at the LGS Group Inc. (now a part of IBM Global Services), and Senior Technical Analyst at MetLife Canada.

He has been co-author, technical editor, or lead author for over 15 books and certification guides. He is an experienced instructor and courseware developer. Recently, he was a Contributing Technical Editor for *Cisco Router and Switch Forensics: Investigating and Analyzing Malicious Network Activity*, (ISBN: 978-1-59749-418-2, Syngress), and *Cisco CCNA/CCENT Exam 640-802, 640-822, 640-816 Preparation Kit*, (ISBN: 978-1-59749-306-2, Syngress).

TECHNICAL EDITOR

Kevin Riggins (CISSP, CCNA) is a Senior Information Security Analyst with over 20 years' experience in information technology and over 10 years' experience in information security. Kevin has used and managed Linux systems since 1995. Kevin has technical and strategic experience in a broad range of technologies and systems. Kevin currently leads the Security Review and Consulting team at Principal Financial Group which performs information security risk assessments and provides information security consulting services for all business units of The Principal. He holds a B.A. in Computer Science from Simpson College, Indianola, IA, is a member of ISSA, Infragard, and is the author of the Infosec Ramblings blog.

CONTRIBUTING AUTHORS

Chris Happel has over 20 years' experience with voice and data networking and security. He is currently a managing consultant for Liberty Trio, LLC, and is an avid supporter of GNU/Linux and open source software.

Terrence V. Lillard (Linux+, CISSP) is an IT Security architect and cybercrime and cyberforensics expert. He is actively involved in computer, intrusion, network, and steganography cybercrime and cyberforensics cases, including investigations, security audits, and assessments both nationally and internationally. Terrence has testified in U.S. District Court as a Computer Forensics/Security Expert Witness. He has designed and implemented security architectures for various government, military, and multi-national corporations. Terrence's background includes positions as Principal Consultant at Microsoft, the IT Security Operations Manager for the District of Columbia's government IT Security Team, and Instructor at the Defense Cyber Crime Center's (DC3) Computer Investigation Training Academy program. He has taught IT security and cybercrime/cyberforensics at the undergraduate and graduate level.

He holds a B.S. in Electrical Engineering, Master of Business Administration (MBA), and is currently pursuing a Ph.D. in Information Security.

Graham Speake (CISSP #56073, M.Inst. ISP) is a risk management consultant with BP, one of the world's largest energy companies. He currently provides risk assessment and remediation consultancy to BP operating units throughout the world. His specialties include industrial automation and process control security, penetration testing, network security, and network design. Graham is a frequent speaker at security conferences and often presents security training to BP staff around the world. Graham's background includes positions as a consultant at ATOS/Origin and an engineer at the Ford Motor Company.

Graham holds a Bachelor's Degree from the Swansea University in Wales and is a member of the ISA. Graham was born in the United Kingdom, but now lives in Houston, Texas, with his wife, Lorraine.

CHAPTER 1

Introducing Linux+

Exam objectives in this chapter

- Benefits of Certification
- Walk through the Book
- Differences in the New Version of the Exam

INTRODUCTION

As I was designing the outline for this book, I thought carefully about the types of individuals who would be tackling the exam. Who would pursue Linux+ certification? There are those who *want* to get ahead and those who *have* to get ahead.

BENEFITS OF CERTIFICATION

If you are reading this book, perhaps you have looked around at your current situation and have chosen to make an improvement. Perhaps you are seeing Linux+ certification as a key credential to put on your resume to apply for a more challenging position in your organization or a different position in a more challenging area. Maybe you are seeing a change in the technology direction your company is taking or your industry is taking and you want to take advantage of any opportunities that may arise.

For those who *have* to get ahead, we know who you are. You are probably faced with a sort of ultimatum: write this exam or else. You may be in a position where holding the latest version of the Linux+ certification is a mandatory qualification. You wrote the 2004 exam, or an earlier version, and now you have to write the 2009 edition to satisfy this requirement. You may have an offer of a position being dangled in front of you, and the 2009 version of the exam is only standing between you and this delicious new role.

> **Note**
> At the time of writing, CompTIA has stated that the Linux+ exam consists of 98 questions and exam candidates will have 90 minutes to complete the exam. The passing mark for the exam is 675 on a scale of 100 to 900. You should confirm these details as you start to prepare for the exam. The exam's Web site is www.comptia.org/certifications/listed/linux.aspx.

Whether you *want* to write this exam or *have* to write this exam, this book will help you get there. All the authors have been through the certification exam experience many times over and have used a multitude of study methods, including self-study. Because you have chosen this book, you have decided to pursue the self-study route, too, or you may be using the book to augment another method. We expect that this book will fit the bill, regardless of what method you choose to use.

A WALK THROUGH THE BOOK

This is a practical book. It is one thing to have an academic or theoretical knowledge of a subject, but that will only take you so far, especially when it comes to writing an exam. Looking at the breadth of subjects that this exam covers, you will need to understand how to install, configure, secure, use, and troubleshoot Linux, not just know about it. This book is structured to walk you through the implementation of a Linux-based workstation or server (or both, if you have the desire and the available hardware). The sections below describe what we were thinking when we wrote the chapters and how the chapters successively build on each other.

The Approach

All authors have multiple vendor-specific and vendor-neutral certifications and all agree that the best way to learn Linux (or any other technology) is to get your hands dirty by installing and configuring it and by using it as often

as possible. This book will walk the prospective exam candidate through the installation and configuration of Linux for use on a server or workstation. To help with this, on the Syngress Web site for the book, we have included the URL to download the Linux distribution and all of us agreed to use it when writing the book. We cannot recommend strongly enough that you download Linux and install it and use it as often as possible. In addition, there are at least three exercises in each chapter to walk you through a myriad of tasks. If you need a hand with the exercises, one exercise from each chapter is available in a guided video on the companion digital video disc (DVD) that is included with the book.

Furthermore, all the authors have taken the latest version of the Linux+ exam and some have taken an earlier version. Speaking plainly, we feel your pain. It makes little sense to tell someone how to write the exam if the author has no idea of what the exam candidate will experience. We recognize that practice exam questions are essential for measuring knowledge. The flavor and type of exam questions had a significant influence on what content was included in the book and how the content was written. Each chapter has 15 exam questions that were written by the authors; therefore, there are 150 exam questions in this book. Should you think that these 150 questions are not enough, there are two full exams on the DVD.

The Chapters

The book has been designed in such a way that you will start with installing Linux and end up with a useable and secure Linux workstation and server that is supported and managed. As it stands right now, there is probably a book, or even many books, that covers every subject that is presented in this one. For example, a simple search on a leading bookseller's Web site with the string "samba linux" turned up 402 books in the Computers and Internet category. This book covers the required content as specified in CompTIA's exam objectives and has been shaped according to the respective exam experiences of the authors. Careful attention has been paid to ensure that each exam objective has been covered and that each term in the list at the end of the objectives has been included in a glossary at the end of the book.

> **Note**
> ComTIA's official exam objectives can be downloaded by requesting them at www.comptia.org/certifications/testprep/examobjectives.aspx. You will need to register first before proceeding to the download page.

The book begins in Chapter 2, Installing Linux, with a walk through the installation process. Hardware compatibility is critical for a successful installation, although it is not the hassle that it once was because most hardware is supported in Linux now. The various *local* and *across the wire* installation methods are also discussed in detail. Laying out the filesystem is a crucial aspect of the installation process, and the various hardware and volume options are described. Once you actually have a filesystem, it needs to be maintained. Because of the breadth of the topic, managing the filesystem merits a chapter unto itself. In this chapter, the types of available filesystems, both local and network, are described as are the tools to use, manage, and repair them once they have been installed.

The hope is that, once the installation process has been completed successfully, your Linux system actually boots up. It is always nice when that happens. Chapter 4, Booting Linux, covers GRUB (the GRand Unified Bootloader), the *de facto* standard Linux bootloader. GRUB can be configured to boot Linux using a number of different kernels or even other operating systems. Knowing GRUB configuration files and commands is important for the exam. Chapter 4 also describes the concept of runlevels and the `init` command, as well as how to troubleshoot boot problems.

With a system that has booted, you can now start to tune and configure your base system, which is described in Chapter 4. This involves creating user profiles and establishing system and environment variables. It also involves configuring additional devices and hardware and establishing network connectivity. It is difficult to imagine any computer these days that is not connected to a network.

Chapter 6, Using BASH, is a key chapter in this book because it covers how to use the *BASH* command-line interpreter (CLI) or BASH shell. A myriad of activities using `bash` commands are described, including directory navigation, file management, file editing with vi, process management, I/O redirection, using special devices, and accessing online help through system documentation. The Linux kernel, the core of the operating system, is also described as are basics of scripting, shell features, automating routine tasks, and managing services.

Note

As you are preparing your system, we recommend that you use BASH whenever possible and avoid using graphical user interface (GUI)-based tools. Working from the command line will give you a better understanding of how to perform system configuration and maintenance and what is happening "under the covers." Furthermore, the exam will only ask you about the use of `bash` commands, not GUI-based tools. You may even find that you prefer to work from the command line.

If you follow the chapters in sequence, then you will install an operating system that is well laid out and configured, connected to a network and has no real purpose yet. Chapter 7, Installing Applications, is all about how to go about installing applications (hence, the clever chapter title) through a variety of available methods to transform your Linux system into a productive tool. In this chapter, you will learn how to install applications from software packages, using `rpm` for .rpm packages in Red Hat-based distributions and `dpkg` for .deb packages on Debian-based distributions, and from source code.

Chapters 8 and 9 provide an overview of what is involved to configuring Linux as a workstation and as a server, respectively. Although there is much overlap in configuring your system as a workstation or as a server, we had to divide the objectives into two chapters to avoid repetition. For example, configuring printing is the same on a workstation and on a server, but it is only described in Chapter 8. Chapter 8 also covers configuring the X Windows System and a display environment using the KDE and GDM window managers and working with multiple desktops. Chapter 9 covers network services, such as Dynamic Host Configuration Protocol (DHCP), domain name system (DNS), Network Time Protocol (NTP), and Samba; Web services, such as Apache (HTTP), Tomcat, File Transfer Protocol (FTP), and Squid; and application services, such as mail and MySQL.

Chapter 10 is another key chapter. There is a definite security focus to the exam, and this chapter, Securing Linux, brings all the listed security objectives into a single chapter. It begins with managing and monitoring user and group accounts, notably the `bash` commands for creating and modifying users and groups, and managing file permissions, including special permissions. Next, we move onto the basics of SELinux. The key to SELinux is to know its different operational modes and policies. Privilege escalation using `su` and `sudo`, and the required /etc/sudoers, is explained. Third, the selection of security applications and utilities is described, including nmap, Wireshark, NESSUS, Snort, and Tripwire. File integrity is essential to ensure against tampering, which necessitates the use of the following checksum, file verification, and encryption utilities: md5sum, sha1sum, and gpg. For those who cannot always be everywhere at once, remote access is available. CompTIA requires you to know both Secure Shell (SSH) and virtual network computing (VNC). Finally, a selection of *authentication* methods are explained, including Pluggable Authentication Modules (PAM), Lightweight Directory Access Protocol (LDAP), network information system (NIS), Remote Authentication Dial-in User Service (RADIUS), and two-factor authentication.

The final chapter, Chapter 11, is all about the care and feeding of a Linux system. This is arguably the most important thing you will do as a system

administrator and the thing that will keep you the busiest. Your available monitoring tools (`sar`, `top`, `iostat`, and `vmstat`, among others) will help you keep your finger on the pulse of your systems in the hopes of proactively correcting an system ills and avoiding any unexpected downtime. The vast number of available logs and the tools that are available for you to analyze them are essential for troubleshooting when things start to go wrong. Finally, backing up and restoring data is the key to your continued employment. For the exam, you are required to know how to back up or restore files using `tar`, `dump`, and `restore`, as well as synchronize files using `rsync`, create disk images using `dd`, and burn CDs and DVDs using `mkisofs` and `cdrecord`.

DIFFERENCES IN THE NEW VERSION OF THE EXAM

As I said earlier in the chapter, all the authors have taken the latest version of the exam that was developed using the 2009 exam objectives. For those of you who took the 2004 version of the exam or who started preparing for the 2004 version and stopped when you found out that a new version was going to be released, there are important differences between the 2004 and 2009 version. According to CompTIA, the following list described the new objectives that will be covered in the 2009 version of the Linux+ exam:

Application and Services

3.4 Given a scenario, explain the purpose of the following Web-related services: Tomcat, Apache, and Squid

3.5 Troubleshoot Web-related services using the following utilities: Commands: `curl`, `wget`, `ftp`, and `telnet`

3.6 Given a scenario, troubleshoot common FTP problems active versus passive; ASCII versus binary

3.7 Given a scenario, perform the following MySQL administrative tasks: Locate configuration file; starting and stopping; test the connection

3.12 Given a scenario, troubleshoot NTP-related issues /etc/ntp.conf; `ntpdate`; `date`; `ntpq -p`

Networking

4.4 Explain the different DNS record types and the process of DNS resolution Local resolution, TTL/caching, root name servers A, MX, PTR, CNAME, NS, TXT

Security

5.3 Explain the basics of SELinux Running modes: enabled, disabled, and permissive

5.7 Deploy remote access facilities using the following: SSH (secure tunnels, SFTP, X11 forwarding, key generation)

5.8 Explain the methods of authentication: PAM, LDAP, NIS, RADIUS, and two-factor authentication

This list comes directly from CompTIA's Web page that introduces the beta version of the 2009 exam: http://certification.comptia.org/linux/betainfo.aspx.

SUMMARY

The bottom line is that we want you to pass your exam using this book. As stated earlier in this chapter, we have been through it ourselves and did so without the benefit of having a book like this to help us study. With this in mind, we put a book together that we would have wanted to use when we went through our own exam experiences. This book covers all the topics listed in the exam objectives and points you to additional sources of information. We hope that this book also makes you "dangerous" with Linux. If you are new to Linux when you start studying, this book will not make you an expert, but it will arm you with enough knowledge and understanding to make you useful in your job. You will also be able to impress your friends who are anchored to another operating system (that is, if they are easily impressed).

We, the authors, wish you every success with the exam and with your career. We hope that you are able to make the changes you seek and that this book helps you achieve your certification. In addition, as individuals who frequently or daily use and enjoy, and are occasionally frustrated by, Linux, we hope that you continue to find a use for Linux and other open-source software whenever and wherever you can.

This page intentionally left blank

CHAPTER 2

Installing Linux

Exam objectives in this chapter

- A Note about Hardware
- Installing from Local Media
- Installing across the Network
- Laying Out the Filesystem
- Disk Types

UNIQUE TERMS AND DEFINITIONS

- **Logical volume manager (LVM)** LVM is a collection of programs that allow larger physical disks to be reassembled into "logical" disks that can be shrunk or expanded as data needs change.
- **Network file system (NFS)** NFS is a protocol developed by Sun Microsystems that allows a computer to mount a volume that resides on a remote computer and access files from across the network as if they were stored locally.
- **Redundant array of independent disks (RAID)** RAID is a form of technology available to Linux systems that uses your disk subsystem to provide enhanced read/write performance, protection against data lost due to disk failures, or both.

INTRODUCTION

The Linux+ certified professional needs to have a good understanding of the overall Linux system before installation. This knowledge is required to successfully install the Linux operating system. In this chapter, you will learn the information needed to configure the Linux system during the initial installation. The Linux+ exam covers the general fundamentals for the installation of the Linux operating system.

Initial operating system installation requires the Linux+ professional understand several important concepts such as computer hardware, system environment settings, partitions, filesystems, and network settings. The Linux operating system's successful installation is predicated upon good planning performed in advance. The planning can be performed formally or informally. However, because of the complexity and flexibility of the Linux operating system process and various different types of computer hardware available, you should plan your installation. This means that you need to determine the role of the Linux system in your environment, gather computer hardware information, and obtain network configuration information so that you can answer questions asked during the installation process.

A NOTE ABOUT HARDWARE

The success of any operating system is dependent on its current and future relationship with hardware vendors. In some cases, operating system vendors also manufacture the system hardware for their own operating system (for example, Apple Macintosh). This approach was very common in the past (for example, Digital Equipment VAX, IBM PS/2, Sun Microsystems Sun/SPARC Family, AT&T 3B2). For those types of operating system vendors, supporting various hardware devices presented several compatibility challenges.

Many of today's major operating system vendors do not manufacture their own system hardware (for example, Microsoft, Novell, and Red Hat). Instead, they have decided to rely on hardware standards used throughout the computer industry. Linux distribution is no exception. Today's Linux operating systems have achieved popularity and powerfulness because its users are benefiting from a wide variety hardware standards used throughout the computer industry. This flexibility is based on the Linux hardware compatibility architecture and the use of the open-source community. The Linux hardware compatibility architecture is divided into four categories, as shown in Figure 2.1.

The four categories are listed below:

- Central processing unit (CPU) architectures supported by the Linux operating system, probably one of the most impressive, continue to

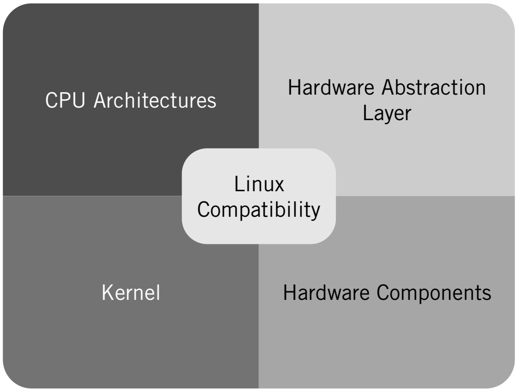


FIGURE 2.1 *Linux compatibility architecture for hardware.*

expand. Some of the CPUs currently supported are AMD, Alpha, ARM, IA-64, m68k, MIPS, PA-RISC, PowerPC (for example, Apple, IBM), S/390, SPARC, 32-bit PC-x86-based (for example, AMD, Intel), and 64-bit PC-x86-64 (for example, AMD, Intel).

- Hardware abstraction layer (HAL) is designed to function as a tier between the physical hardware and the software functioning on the system. As an abstraction layer, its purpose is to hide hardware complexity and differences from the operating system kernel. This approach allows you to select a CPU-specific Linux kernel for functioning on different hardware systems. Today's modern operating systems (for example, Windows, BSD, Linux, MAC OS X, CP/M, Solaris) are designed to interoperate with the HAL.
- Linux monolithic kernel is an architecture designed to function dynamically. It supports the loading of modules and instructions to implement all operating system services (for example, process management, concurrency, and memory management). The Linux kernel integrates the CPU architecture via a series of device drivers and kernel extensions.
- Hardware components are presented as device drivers (for example, printers, monitors, video cards, storage devices, and modems) within the Linux environment. For each Linux version (for example, openSUSE and Red Hat), a unique Linux Hardware Compatibility List is created. The Linux Hardware Compatibility List contains supported hardware devices for that specific Linux version.

The overall four-category approach of the Linux operating system extends its functionality. This approach to modularize the Linux Hardware Architecture allows it to function with an extremely wide range of computer hardware components. In fact, this approach has enabled Linux to work with

every conceivable piece of hardware. In addition, the decision to support the open-source community has also played a critical role in extending the Linux hardware environment. The open-source environment continues to add new hardware components daily through the development and support of various Linux hardware components. The hardware open-source program is a part of the open source culture. For the four categories, several Web sites provide a list of compatible systems and devices. The Linux community uses several Web sites, including the following, to determine the compatibility of various hardware and full systems supported by various Linux distributions:

- www.tldp.org/howto/hardware-howto
- http://en.opensuse.org/hardware
- www.linux-drivers.org
- http://hardware4linux.info
- www.linux.org/hardware

Learn By Example: Measure Twice, You Can Only Cut Once

During my early years as a consultant, I was requested to provide an emergency support to a large corporate customer. I was contacted on Saturday and requested to show up first thing Monday morning. The large customer migrated all of their main servers supporting a critical application from one operating system to a different operating system. I was informed that the migration went well and that all the applications on the servers worked fantastically. This included the major applications. The only problem was that the customer and their user-supported community could not print. No printing! This included not being able to print invoices or payroll checks from the main application. As a result, they needed emergency services to determine why this was the case and what would be their next steps. Upon arrival, I reviewed the customer environment and sized up the magnitude of the printing problem.

Based upon my ace superhero skills, I was able to ascertain that no print drivers existed for the 20 special printers within the new operating system that the customer migrated toward. Their mission critical application could not print because there were no print drivers. In addition, the customer could not go back to the previous operating system servers. Those servers were reformatted and rebuilt to support the new operating system. Migrations are typically one way. . .forward! The customer terminated their contract with the previous consulting firm and hired my team to resolve the current situation and assist them with the migration of the remaining servers.

Do you ask, how did I know so quickly? I checked the new operating system vendor's Hardware Compatibility List (HCL) and also with the printer vendor's technical support team. The HCL is very important! Remember software migration is not the only issue, you may also be required to migrate peripheral devices (for example, printers

and modems). I was able to get the customer to print from the existing printers. How? I noticed on the HCL a printer mode that the existing 20 printers could emulate. This approach was used until the printer vendor was able to develop new print drivers for the customer's new operating system. I miss the good old days!

The knowledge of the four-category approach for the most modern operating systems is a requirement for any computer professional. The Linux professional is no exception. Ironically, with the development of kernel-based architectures, many of today's operating systems hide a lot of their computer hardware functionality within the operating system. The Linux operating system does not. As a result, you should have a good understanding of how the Linux computer hardware works. For the Linux+ exam, you will need a fundamental understanding of the following components:

- Power supplies are the devices required to provide the various computer hardware components (for example, motherboards and internal disk drives) within your system with direct current (DC). The power supply regulates the alternating current (AC) received from a wall's outlet by transferring it into DC required by computer chips; it is housed inside the system unit.
- Motherboards are perforated circuit boards housed inside the system unit that contains the CPU, the memory (RAM) chips, and the slots available for the expansion cards. It is the largest printed circuit board; all the other expansion boards (for example, video cards and sound cards) interface with it to receive power and to provide bidirectional communications.
- CPUs, the brains of the computer, are responsible for data processing and are the most important chip in the computer. CPUs control the functions performed by the various hardware components, processes all software instructions issued, and determines the speed of the system. The CPU is housed inside the system unit.
- Memory, implemented as computer chips, helps process data or instructions by storing the instructions or data that the CPU processes.
- Expansion boards are devices that expand the capabilities of a computer by enabling a range of different devices (for example, monitors, speakers, and modems) to communicate with the motherboard.
- Video adapters are expansion cards that translate binary into the images viewed on the computer monitor.

- Storage devices are internal and external devices (for example, hard disk drives, floppy disk drives, and CD drives) used for storing data and information.

> **Note**
> The Linux+ certification exam will not test you on computer hardware. However, it is important that you have a fundamental knowledge of the overall Linux Hardware Architecture.

For the Linux professional, the selection of which Linux distribution to use is dependent on the four hardware categories presented above. After selecting the desired Linux distribution, most distributions will allow you to install and configure your system hardware components during the installation phase. In addition, most distributions also support the installing, configuring, updating, and removing of hardware components after the system has been installed.

EXERCISE 2.1: OpenSUSE Linux Hardware Compatibility List

In this exercise, we will review the Linux Hardware Compatibility List to determine whether openSUSE 11.1 can be installed on the Dell PowerEdge PE-T605 and HP Proliant DL380 G3 Servers. Complete the following:

1. Open a **Web browser** from a workstation and navigate to http://en.opensuse.org/hardware.
2. The compatible hardware can be viewed for full systems or individual components. Select the **Server category** underneath the Full Systems section. Is the Dell PowerEdge PE-T605 supported? Are there any installation problems listed?
3. How about the HP Proliant DL380 G3? ■

INSTALLING FROM LOCAL MEDIA

The Linux operating system's successful installation is predicated upon good planning performed in advance. The purpose of this section is to present you with the major Linux installation decisions required during the operating system installation process. The decisions, presented within screenshots, will display the basic installation of the Linux operating system. The screenshots present the options you will face during the installation of the Linux operating system.

Linux Installation Process

The Linux operating system can be installed from various different local [for example, CD, digital video disk (DVD), and iso images] and network [for example, File Transfer Protocol (FTP), Hypertext Transfer Protocol (HTTP), NFS, and Server Message Block (SMB) Server] sources. The *local media installation* method entails performing a stand-alone installation and does not rely on network connectivity to access to the installation source data. The *network source installation* method entails having an installation server available in your network or via an external server (for example, Internet) as the source where the installation data resides. Apart from a few unique settings, the installation process for each installation source is essentially the same. The "Installing across the Network" section presents the unique challenges associated with the installation of a Linux distribution from network sources. This section presents the installation of the Linux distribution from local media. Linux local media installation can be performed by using CD, DVD, or .iso sources. Of the choices provided, the author prefers .iso image installations.

As there are several Linux distributions available from various vendors and some are more popular than others, the first choice you face is the selection of a Linux distribution. The leading Linux distributions all follow the same installation stages. Gentoo is a notable exception. For this book, the authors have decided to use the openSUSE 11.1 Linux distribution. OpenSUSE 11.1 is the most current version available.

> **Note**
> The Linux+ certification exam is a vendor neutral exam. For a listing of the various Linux distributions, check out the following Web sites:
>
> - http://distrowatch.com/
> - http://en.wikipedia.org/wiki/List_of_Linux_distributions

EXERCISE 2.2: Installing from Local Media

In this exercise, we will commence the initial installation of the openSUSE 11.1 installation using DVD Media. For this exercise, we will use the following:

1. Open a **Web browser** from a workstation and navigate to http://software.opensuse.org/.
2. Follow the instructions to download the openSUSE 11.1 DVD image.

3. Create a bootable DVD from the .iso image using your burning software application.
4. Boot the target machine with the newly created bootable network DVD.
5. Select **Installation**. This will start the openSUSE Installer. ■

To launch the installation process from an .iso image, load the image as a CD/DVD into the virtualization software (for example, VMWARE workstation). Power up or reboot the virtual machine with the .iso image loaded to commence the initial boot process. The .iso image appeared as a DVD to the virtual machine. When the system completes the on-board self test, the openSUSE Installer program screen appears, as shown in Figure 2.2. The initial screen provides you with various menu options for installing openSUSE and additional configuration options required before the installation process. We will select the **Installation** option.

Below are descriptions of the choices:

- **Boot from the hard disk** This option is used to automatically boot an existing Linux operating system (if previously installed on the hard disk). This prevents the automatic installation of a new Linux operating system.

FIGURE 2.2
OpenSUSE Installer boot screen.

- **Repair installed system** This option repairs a previously installed system. It boots the graphical repair system. This option also appears again after the System Probing phase covered later in the chapter.
- **Installation** This option loads the mini Linux kernel from the Linux distribution and starts the installation process.
- **Rescue System** This option starts a specialized small Linux kernel without a graphical user interface. It loads the Linux kernel into RAM and can modify configuration files, check the filesystem for defects, verify and/or modify the boot loader configuration, resize the partition, and a few other critical system modifications that may be necessary.
- **Check installation media** To verify you have an authentic copy of Linux before installing, this option checks .iso image's integrity. It uses the MD5 cryptographic hash algorithm. To verify the integrity of other media, a number of free MD5 and SHA cryptographic hash algorithms are available and downloadable from the Internet.
- **Firmware test** This option tests the system's Basic Input/Output System (BIOS) and Advanced Configuration and Power Interface (ACPI) compatibilities.
- **Memory test** This option conducts systematic tests of your system RAM using memtest86. The memtest86 program is a stand-alone application for testing the physical memory on x86-based systems.

The remaining options located in the bar at the bottom of your screen are used as follows:

- **F1 – Help** The Function 1 Key (**F1**) provides helpful information about the various options.
- **F2 – Language** The key allows you to select a different language for the installation of Linux (The default language is English).
- **F3 – Video mode** The key allows you to select a desired graphical (screen resolution) or text mode for installation.
- **F4 – Source** This selection determines the Linux installation source (for example, FTP, HTTP, NFS, CD, DVD, SLP), as shown in Figure 2.3. The "Installing across the Network" section presents this option in greater detail. The default installation source selected is DVD.
- **F5 – Kernel** The key allows you to disable potential hardware components (for example, ACPI systems and DMA modes).
- **F6 – Driver** This key installs optional drivers or updates existing drivers.

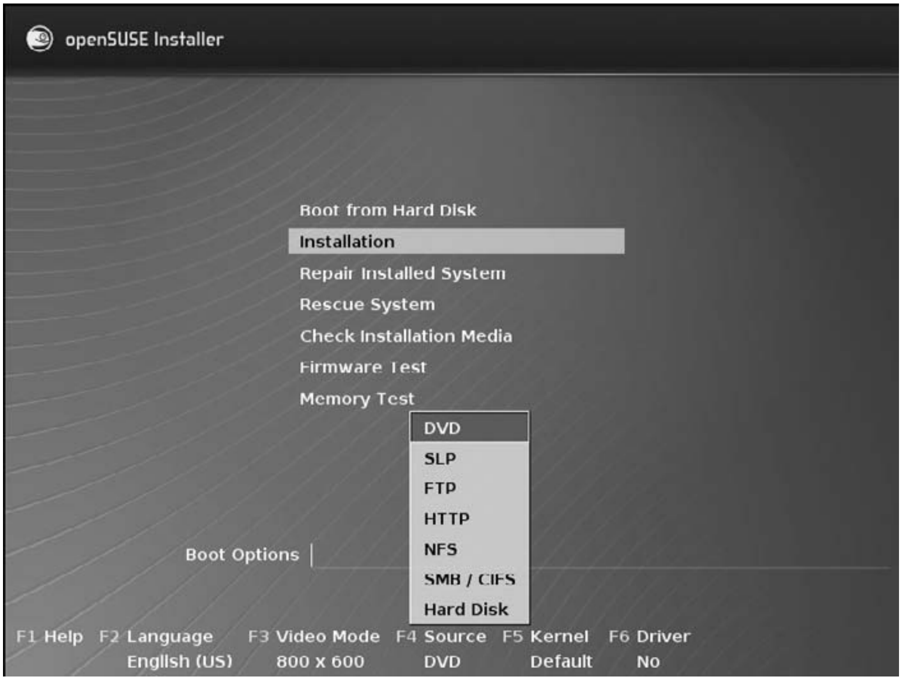


FIGURE 2.3 *Network installation source options.*

The openSUSE installation process is divided into three distinct stages for ensuring you are successful in building a Linux system. The three stages are preparation, installation, and configuration. The preparation stage assists you in configuring the system's language, data and time, desktop environment, user account information, user and root password authentication methods, and type of disk partitioning information. The installation stage is a noninteractive process. This stage installs the openSUSE software and prepares your system for the initial boot sequence. The final stage is the configuration stage. During this stage, depending on whether you select automatic or manual configuration, the installer software will either preconfigure various network settings or allow you to input network configuration information. For example, your machine's host name, domain name, and various other network configuration settings [that is interfaces, Dynamic Host Configuration Protocol (DHCP), and firewalls].

WELCOME SCREEN

The first stage, preparation, collects information from you regarding your system's environment and your preferences. The openSUSE installation

FIGURE 2.4
Welcome screen for keyboard, language, and license agreement settings.

procedures start by displaying a Welcome screen, as shown in Figure 2.4, allowing you to select a language and read and agree to the terms in the License Agreement.

The default language and keyboard settings are English (US). For those wishing to install Linux in a different language, a plethora of language and keyboard options are available. This wide variety of languages is one of the nice benefits of an international operating system. Selecting the language and keyboard settings will automatically switch the system to the prescribed settings. Next, you should read the License Agreement throughout its entirety. If you agree with the terms, check the I Agree to the License Terms. If you disagree with the license agreement, openSUSE will not be installed.

System Probing and Installation Mode

The openSUSE Installer application performs a system analysis of your system by conducting a system probe, as shown in Figure 2.5, to search for various storage devices (for example, USB, Firewire, floppy disks, and hard disk drives), existing Linux partitions and system files, determining whether the system can be updated, and launching the Package Manager.

FIGURE 2.5
System probing.

The next screen displays the various installation modes after the system analysis is completed, as shown in Figure 2.6a. The automation configuration selection is enabled by default. This is the preferred selection for performing a new installation when default hardware and network settings can be used. The various hardware and network settings can be changed after the system is installed.

During the Installation Mode, you have the following choices:

- **New installation** This option installs a new copy of the Linux operating system.
- **Update** This option performs an update of a previous openSUSE installation.
- **Repair installed system** This option repairs a damaged version of a previously installed Linux system.
- **Include add-on products from separate media** This option installs additional products. These can include support for additional languages and third-party products. If this option is selected, Add-On products can be installed from a network or local source. If you select a network source, you must configure the network settings and install a network card.

FIGURE 2.6
(a) Installation mode with settings for automatic configuration.
(b) Installation mode with granular settings for configuration stage.

- **Use automatic configuration** This feature will attempt to automatically configure most network and hardware configuration settings. If necessary, changes to the various settings performed during this stage can be made after the installation process.

If you prefer to provide customized hardware and network configuration information during the installation process, you can uncheck the **Use Automatic Configuration** setting and provide the necessary information later on during the configuration stage. Figure 2.6b displays a different Installation Mode screen with more granular settings to be performed during the configuration stage. This modified screen will appear when the **Use Automatic Configuration** setting is unchecked.

Clock and Time Zone

The Clock and Time Zone screen allows you to set the Region, Time Zone, and system Date and Time information, as shown in Figure 2.7. In addition, you can determine whether you would like to use local time or coordinated universal time (UTC). Another name for UTC is Greenwich Mean Time (GMT). The system can be configured to use Network Time Protocol (NTP)

FIGURE 2.7
Clock and Time Zone.

after the installation process is completed, if the network is not already configured. The openSUSE Installer allows you to use the mouse to select a geographical area on the map and zoom in closer to select a country or a region.

Desktop Selection

The Desktop Selection screen allows you a choice between various user interface options, as shown in Figure 2.8. **GNOME** *desktop environment* and *K desktop environment* (**KDE**) are the most popular interfaces. In addition, you can select the **other** option. The **other** option, normally used on servers and selected appliances, provides a minimal graphical environment or text-mode only installation.

Both the KDE and the GNOME desktop environment are free software programs. They both provide a desktop environment for users and an extensive development framework for developers. In addition, both function in various languages. Both interfaces provide a unique look and feel, and selecting one over the other is a matter of style and preference. The current interface versions implemented in openSUSE 11.1 are GNOME 2.24 and KDE 4.1. The authors elected to use KDE 4.1 for this book.

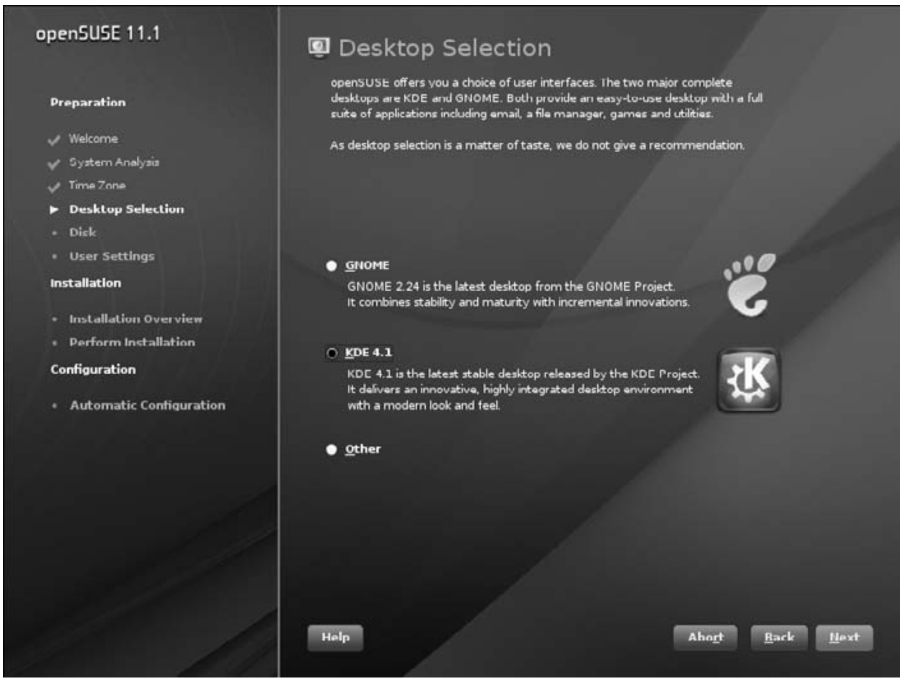


FIGURE 2.8
Desktop selection – user interfaces.

Suggested Partitioning

Suggested partitioning leads you through the process of selecting and implementing a partition and filesystem schema based upon your disk layout. If possible, openSUSE Installer provides you with a recommended partition setting, as shown in Figure 2.9. For most new installations and first time users, the installation default will be acceptable. The next screen will present you with the choice to select the default **partition-based** option or the *logical volume management* (**LVM**)-*based* option. The default selection is the **partition-based** option. In addition, the user can choose to edit the existing partitions or create new partitions. The decision to create a new partition or edit an existing partition depends on whether the Linux system will coexist with an existing operating system (for example, Microsoft Windows XP), which contains more than one disk drive, or whether you want to resize a foreign filesystem partition (for example, NTFS). For additional information on this process, consult the openSUSE online documentation.

The recommended partitioning schema represents the most common approach. This approach entails having two primary partitions and one extended partition. The two primary partitions support the Linux *root* partition and the Linux *swap* partition. The extended partition supports the

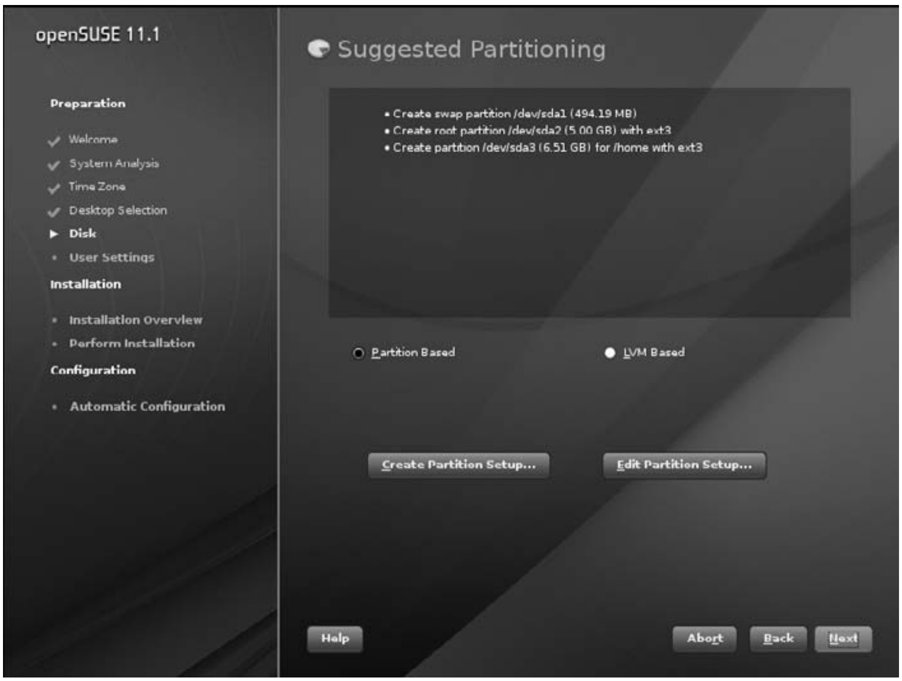


FIGURE 2.9
Suggested partition.

home partition. The "Laying Out the Filesystems" and "Disk Types" sections provide detailed information regarding disk partitions, filesystems, LVM, and RAID. At present, we will accept the default partition.

User Settings

The User Settings screen creates the Linux user account, a password for the root account, and selects either local or network authentication, as shown in Figure 2.10. During this process, you must provide the following information:

- **User's Full Name** The user's first name and surname (full name) must be entered.
- **Username** The username for logging in to the system must be entered.
- **Password** The password grants access to the system. Passwords can be alphanumeric and are case sensitive. They should not contain any accented characters or umlauts. If you enter a password that is easy to guess (for example, dictionary word, user's last name), the system will provide you with a warning. This occurs because the system automatically checks for weak passwords.

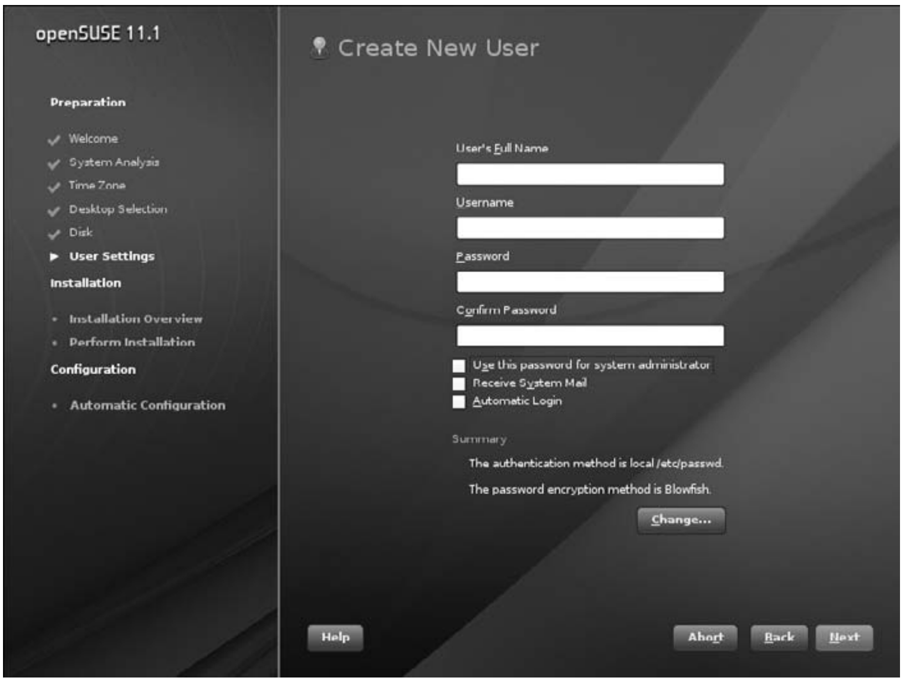


FIGURE 2.10
Create user screen.

- **Confirm password** The same password as entered above is reentered for confirmation.

The screen also provides you with the option to assign the same password to the system administrator "root" account. For better security protection, it is best not to use the same password for both accounts. The user account can receive system errors, install package updates, and other critical messages normally made available for the system administrator by selecting the **Receive System Mail** option. The local mailbox, /var/spool/mail/username, where username is the login name of the selected user, is the directory in which mail messages are stored. The next option offers you the ability to set the **Automatic Login** feature to allow the system to automatically log you into the system whenever the system restarts (or reboots). For security reasons, this option should not be selected. It is necessary for Linux users to enter their usernames and passwords just in case they decide to store sensitive information on the system.

Finally, you will be given the option to change your authentication and password encryption methods, as shown in Figure 2.11. The various authentication methods are as follows:

FIGURE 2.11
Authentication and password encryption methods.

- **Local** This option inserts the user account information into the /etc/passwd file. This is the typical default configuration for most Linux systems.
- **Lightweight Directory Access Protocol (LDAP)** OpenLDAP provides a directory service for the storing of user accounts within a hierarchical database. This approach allows the database to replicate the user account information with other OpenLDAP systems
- **Network information system (NIS)** NIS (originally known as *Yellow Pages* or YP) is a distributed account management system based on a client-server directory service protocol model. NIS communicates account information between Linux systems.
- **Windows domain** Windows Domain authenticates user account information with Microsoft Active Directory.
- **Kerberos** Kerberos is a symmetric cryptographic protocol for network-based authentication.

The password encryption methods available to you are as follows:

- **Blowfish** This is a cryptographic symmetric block cipher used to encrypt the password. The password length for the *Blowfish* encryption method ranges from 5 to 72 alphanumeric characters.
- **MD5** This is a cryptographic hash function used to create a hash value for the password. The password length for the *MD5 algorithm* method ranges from 5 to 127 alphanumeric characters.
- **Data encryption standard (DES)** DES is a cryptographic symmetric block cipher standard for encrypting your password. The password length for the *DES encryption* method ranges from five to eight alphanumeric characters.

Installation Settings

The second stage, installation, presents you the collected installation settings. This allows you to review and modify, if desired, any of your settings or any openSUSE Installer system default configuration suggestions, as shown in Figure 2.12. During this stage, you have the option to accept the recommended installation settings or make the necessary modifications.

You can change the disk partition and filesystem setup, the boot loader default boot sequence (for example, GRUB), the default software packages, the adding of additional software packages, the Language and Keyboard

FIGURE 2.12
Installation settings confirmation screen.

Layout settings, the user and root account and password authentication method settings, the defaulted boot runlevels, hardware components and settings, and third-party software image installations. Chapter 4 presents the Linux runlevels in detail.

Perform Installation

The perform installation screen indicates the system will start the installation process. During this process, the openSUSE Installer presents you with three different tabs. The first tab, **Slideshow**, shows you information about Linux features, as shown in Figure 2.13a. During the slideshow for this distribution, information pertaining to how to keep your system up-to-date, the default installation of a firewall and other security concerns, and the location of Linux is presented.

You can monitor the installation of the operating system by selecting the **Details** tab, as shown in Figure 2.13b. This includes the creation of the partitions, the formatting of the partitions, the filesystems assigned to each partition, and the mounting of the partitions. Finally, you can review the updated information that is not included in manuals by selecting the **Release Notes** tab.

FIGURE 2.13
(a) Linux installation slideshow. (b) Linux installation details.

FIGURE 2.14
Preparation for initial boot.

The last step in the installation stage prepares your system to boot into the new Linux operating system, as shown in Figure 2.14. The finishing basic installation procedures include copying the system files to the system, saving any system configurations, installing the Boot Manager, saving any installation settings, and finally, preparing the system for the initial boot.

Automatic Configuration

The automatic configuration commences the configuration of your system's default settings, as shown in Figure 2.15. This occurs during the configuration stage. During this process, hardware configurations, network configurations [for example, host and domain name, network card(s), and DNS], and any other services will be automatically set up.

Manual Configuration

If you did not select the **Automatic Configuration** option earlier, the system restarts during the configuration stage into manual configuration mode. The manual configuration mode allows you to enter hostname, domain name,

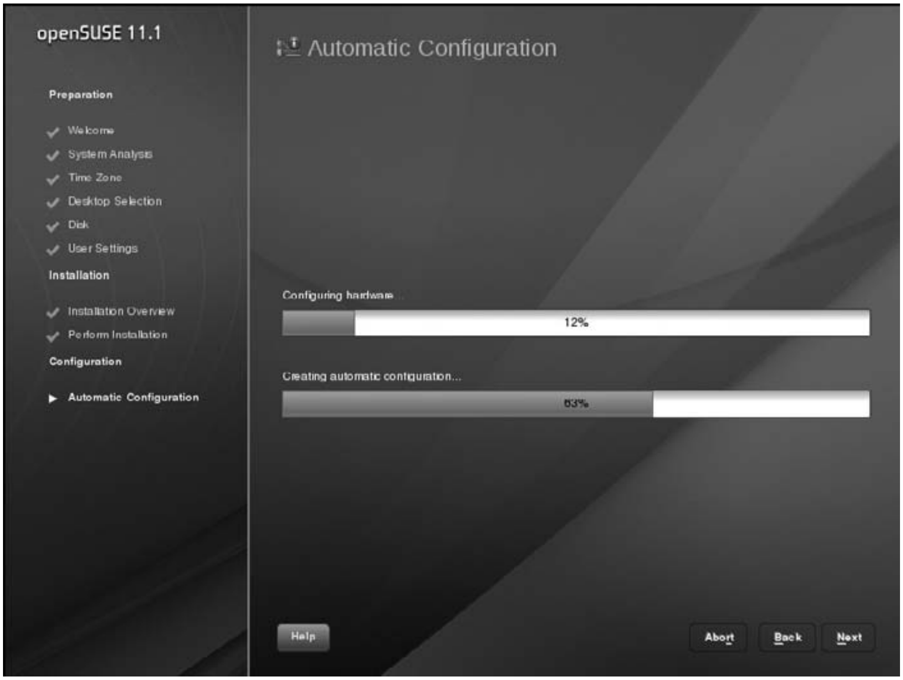


FIGURE 2.15
Automatic configuration process.

and network configurations. In addition, you will be able to test Internet connectivity capabilities, perform online updates, review release notes, and perform additional hardware configurations for your system (for example, printers, sound cards, and graphics cards).

Hostname and Domain Name

The Hostname and Domain Name screen allows you to assign a unique computer name to your system. This unique name is required to participate in a network. Figure 2.16 presents the options for configuration of your hostname and domain name.

Your system's hostname can be manually entered or assigned by checking the **Change Hostname via DHCP** instead option. This is the default setting. The setting **Write Hostname to /etc/hosts** enables your machine to be accessible even when not connected to a network. This default setting should remain enabled. In addition, your system requires a domain name. The domain name, typically a common name, is shared by all hosts on your network.

FIGURE 2.16
Hostname and domain name settings.

Network Configuration

The Network Configuration screen, as shown in Figure 2.17, allows you to configure various network and system security settings (for example, firewall). The screen offers you the opportunity to configure the network and security now or at a later time after the system installation is completed. If you decide to configure the network settings now, you have several options during this phase.

The general network settings will enable or disable the network manager tool for laptop configuration. In addition, it enables or disables the IPv6 support. IPv6 is enabled by default. The firewall settings, enabled by default, allow you to designate the services and ports you want to access via the network. The firewall security settings you configure apply across all configured network interfaces. The network interfaces allow you to set up any newly installed network cards and change existing network interface configurations. For remote access via modems (telephone modems, DSL, or ISDN), the next three options allow you to configure the devices via unique configuration dialog boxes. Finally, the screen allows you to configure your system for VNC remote administration and proxy support.

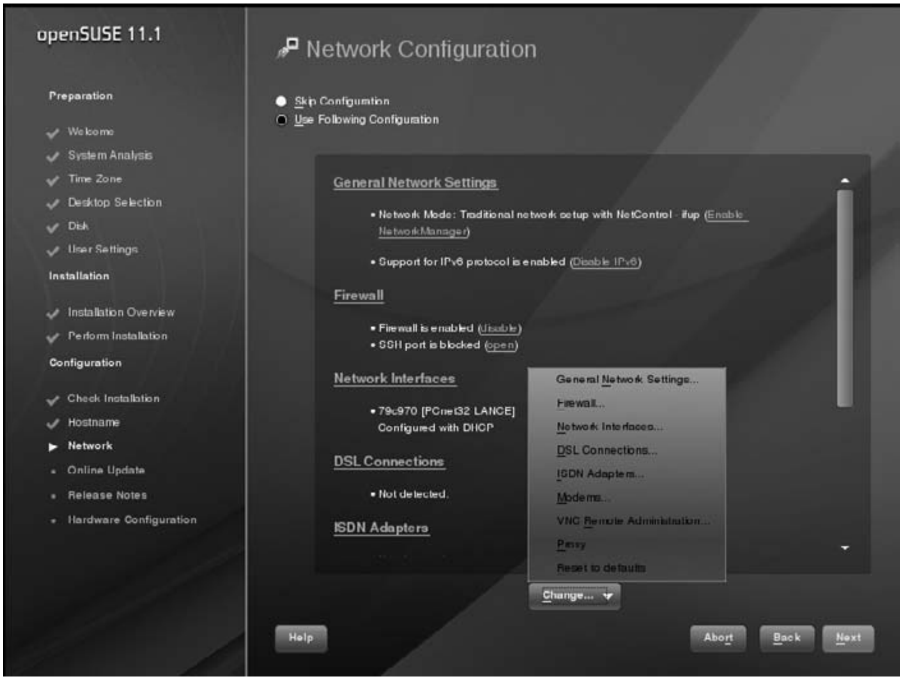


FIGURE 2.17
Network configuration settings.

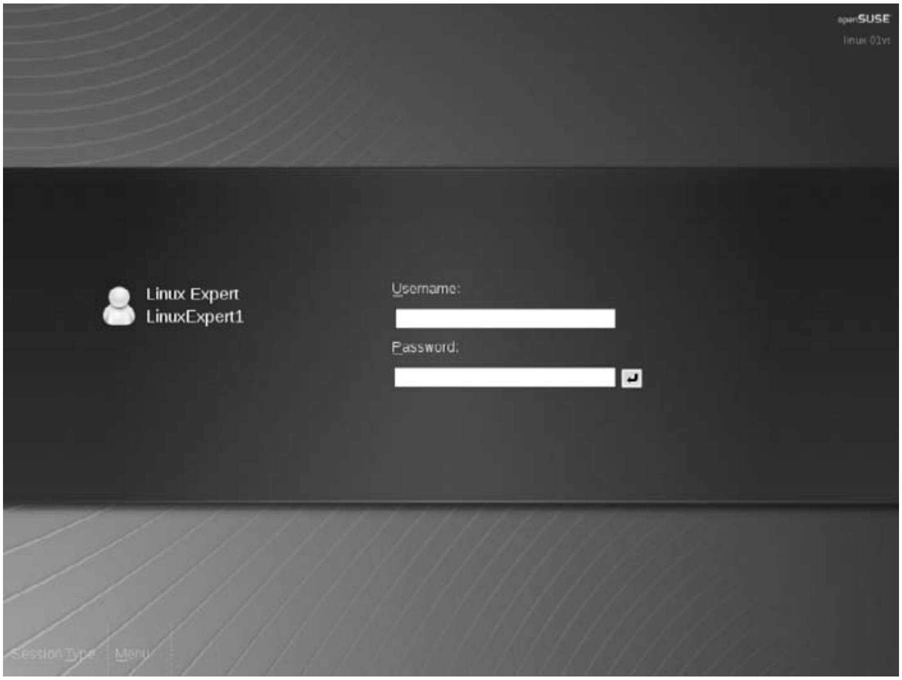
The openSUSE Installer tool allows you to test Internet connectivity and perform several other additional functions (for example, obtaining the latest release notes, registering for technical support, and performing online updates). This includes the option to perform additional hardware configurations and add various other devices (for example, printers) to your system.

Congratulations, the openSUSE installation and configuration process is complete. Your system should now display the graphical login screen on your computer monitor, as shown in Figure 2.18. This screen allows you to enter a username and password to log in to the system.

INSTALLING ACROSS THE NETWORK

Network source installation is an alternative approach to the local media source installation described above for a Linux distribution. This approach offers you both advantages and disadvantages. The major disadvantage is its slowness as compared with the local media approach. The slowness is due to the impact on network bandwidth. Network-based speed is much slower

FIGURE 2.18
Graphic login.



than hard disk drive speeds. The major advantage is that you can install Linux on multiple machines simultaneously. However, this increases impact on network bandwidth.

To install Linux across an internal organization network, the primary network protocols available are HTTP, FTP, NFS, and SMB. In addition to internal network installs, you can install Linux via the Internet using the HTTP and FTP protocols. Before commencing the network-based installation, the server containing the Linux distribution source must be properly configured to support the desired network protocol, and the appropriate Linux distribution must be accessible via either a CD, DVD, or .iso image. The targeted Linux system must be started from a network bootable CD or DVD image. The openSUSE 11.1 network bootable image is called *Mini CD*. You can download it from the openSUSE Web site and burn it to a disk before booting your targeted Linux system.

When the system completes the on-board self test and loads the mini Linux kernel, the openSUSE Installer boot option screen, as shown in Figure 2.19, will appear providing you with various menu options for installing openSUSE and additional configuration options required before

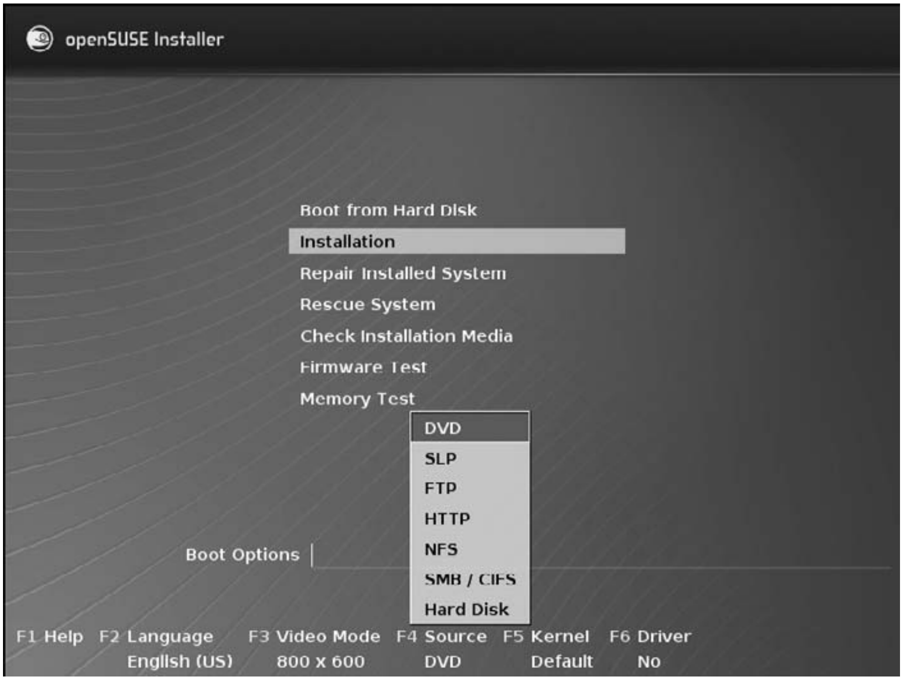


FIGURE 2.19
Network installation source options.

the installation process. From this screen, you select the desired network installation source (for example, DVD, SLP, FTP, HTTP, SMB, and Hard Disk) by pressing the **F4** Key. For the Linux+ certification exam, we will concentrate on the HTTP, FTP, and NFS protocols. Sections "A Note about Hardware" and "Installing from Local Media" present this option.

For HTTP-based network installations, you enter the server containing the Linux distribution IP address or domain name, as shown in Figure 2.20. The server's IP address or domain name can represent a local server or a server located on the Internet. In addition, you enter the Linux distribution source directory or folder.

EXERCISE 2.3: Installing across the Network

In this exercise, we will commence the initial installation of the openSUSE 11.1 installation across the network using the HTTP installation procedures. For this exercise, we will use the following:

- **Server/Domain Name:** http://download.opensuse.org
- **Directory/Folder Location:** distribution/11.1/repo/oss/

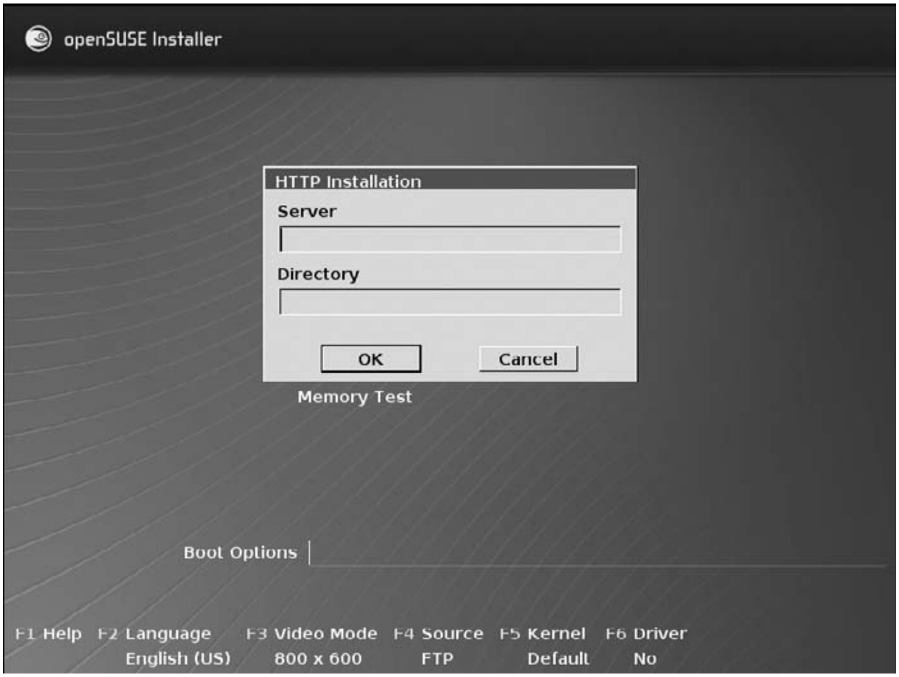


FIGURE 2.20 *HTTP network installation.*

Complete the following:

1. Open a **Web browser** from a workstation and navigate to http://software.opensuse.org/.
2. Follow the instructions to download the openSUSE 11.1 network installation Boot CD.
3. Create a bootable CD from the .iso image using your burning software application.
4. Boot the target machine with the newly created bootable network CD.
5. Press the **F4** key to specify the installation source.
6. Select the **HTTP** option as the installation source.
7. Enter **Server/Domain Name**: http://download.opensuse.org
8. Enter the **Installation Source Directory Location**: distribution/11.1/repo/oss/
9. Select **Installation**. This will start the openSUSE Installer. ■

For the FTP-based network installations, you enter the server containing the Linux distribution IP address or domain name, as shown in Figure 2.21. The server's IP address or domain name can represent a local server or a server located on the Internet. In addition, you enter the Linux distribution source directory or folder. Finally, enter the FTP user account name and password to obtain authentication access. The system can support anonymous access.

For NFS-based network installations, you enter the server containing the Linux distribution IP address or domain name, as shown in Figure 2.22. The server's IP address or domain name can represent a local server or a server located on the Internet. In addition, you enter the Linux distribution source directory or folder.

Table 2.1 summarizes the type of network installations performed and the parameter values you will need to provide during the initial installation.

After selecting the preferred network-based installation method, the openSUSE Installer returns to the main screen presented earlier in the "Installing from Local Media" section. Select the **Installation** option to commence the openSUSE installation process.

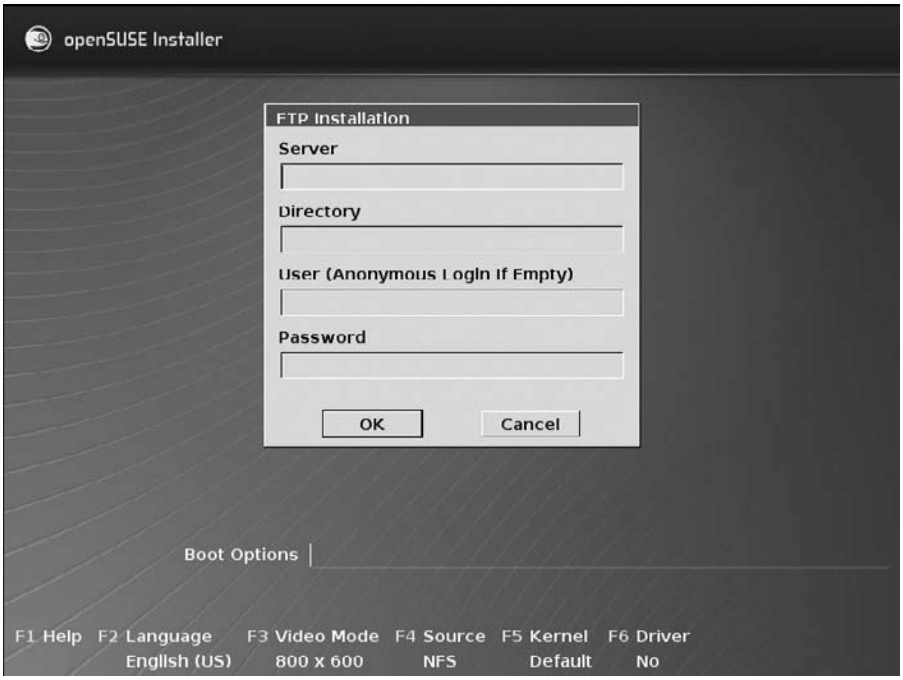


FIGURE 2.21
FTP network installation.

FIGURE 2.22
NFS network installation.

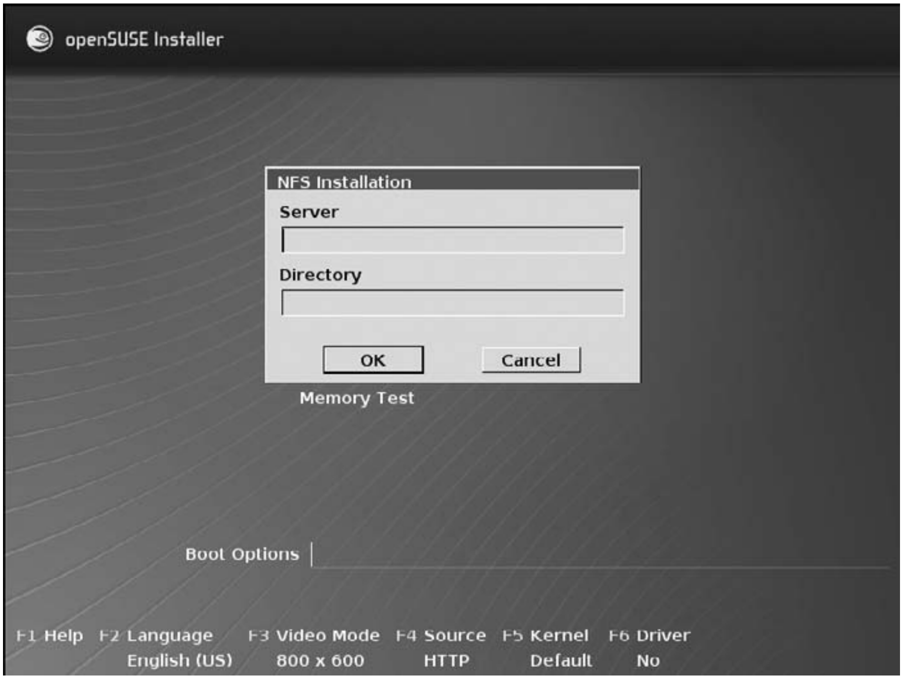


Table 2.1 Network Installation Parameters

Network Installation Type	Domain Name/IP Address	Distribution Source Directory/Folder	User Name	Password
HTTP	Required	Required	N/A	N/A
FTP	Required	Required	Required (anonymous login if empty)	Required
NFS	Required	Required	N/A	N/A

LAYING OUT THE FILESYSTEM

In the "Installing from Local Media" section, the openSUSE Installer provided you with a recommended default partitioning for your system's hard disk drive. Figure 2.23 depicts the recommended partition settings. For this section, instead of you accepting the recommended partition settings, you can choose to edit the proposed existing partitions or create new partitions.

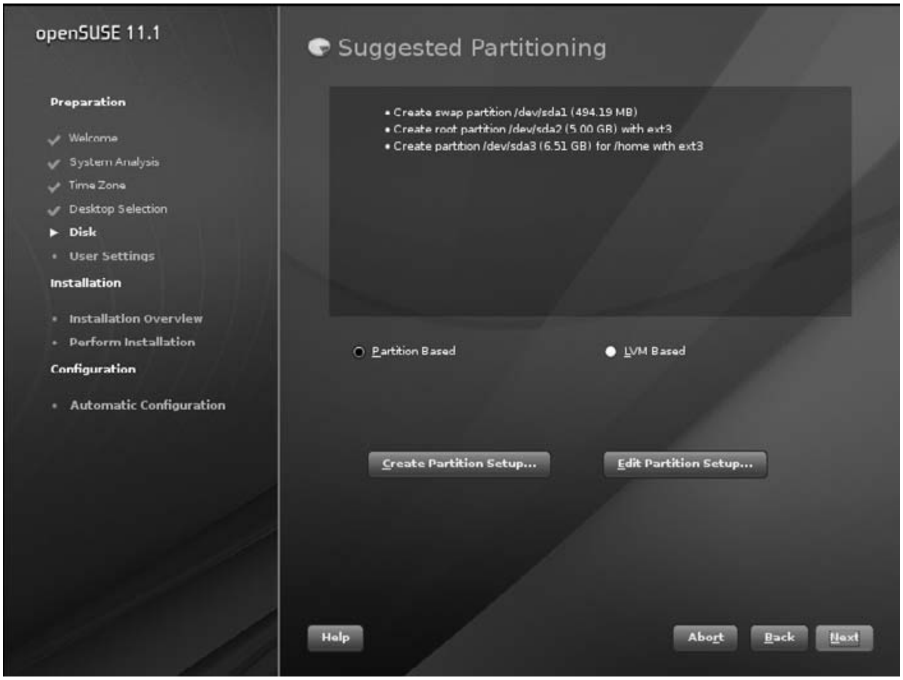


FIGURE 2.23
Suggested partitioning.

This section provides information you can use if you decide not to accept the recommended partitions. OpenSUSE presents you with the choice to select the default **partition-based** option or the **LVM-based** option. The *partition-based* option is the default selection. It will be presented first. The *LVM-based* option will be presented afterwards.

Before we commence selecting any option, some initial knowledge about disks and partitioning is required. A physical hard disk, the kind you can purchase from a store, has several limitations; some are imposed by the disk manufacturer (for example, disk geometry) and others are imposed by your system manufacturer (for example, BIOS). Your hard disk drive is confined to operate within these constraints.

Partition types are the first major constraints for hard disk drives. There are two major categories for partitions. Primary partitions are the first type. This type of partition divides the hard disk drive into physical groups. For most PC-based systems, a maximum of four primary (physical) partitions can be implemented on a hard drive. Extended partitions are the second type. Extended partitions can be further subdivided into smaller logical partitions (groups). By allowing you to further subdivide your extended partition into smaller logical partitions, you can create more partitions on your system.

This approach allows you to go beyond the maximum four primary partition limitations.

For any bootable operating system, you must have at least one primary partition. This initial partition will be used by the operating system to store your operating system files. The remaining three partitions can all be the primary partitions, the extended partitions, or a combination of both. Therefore, partitions are the physical and/or logical dividing of your hard disk drive into one or more partitions. For a dual-boot system, you must have two primary partitions (one for each operating system).

Exam Warning: Primary versus Extended Partitions

Primary partitions divide the hard disk drive into physical groups and cannot be further divided. For most PC-based systems, a maximum of four primary (physical) partitions can be implemented on a hard drive. Extended partitions can be further subdivided into smaller logical partitions (groups). For any bootable operating system, you must have at least one primary partition. This initial partition will be used by the operating system to store your operating system files. The remaining three partitions can all be the primary partitions, the extended partitions, or a combination of both.

To create or edit partitions within the Linux environment during the installation process, openSUSE Installer provides you with the Expert Partitioner graphical interface tool. The decision to create a new or edit an existing partition should be based upon whether the Linux system will coexist with an existing partition, contains more than one disk drive, or whether you want to resize a foreign operating system's partition (for example, NTFS).

Figure 2.24 displays the recommended Expert Partitioner partitioning scheme. It comprises two primary partitions and one extended partition. The two primary partitions are the Linux root partition and the Linux swap partition. The extended partition will be used for the home partition.

Another approach to view, edit, and create partitions is via the use of the `fdisk` command-line tool. The `fdisk -l` (lowercase `l`) command displays all disk drives attached to your system and their corresponding disk geometry, as shown in Figure 2.25. The output from the command shows the disk device name and size, the disk geometry, the disk identifier number, and the existing partition map (if a disk drive is accessible).

To create or modify partitions using `fdisk`, you must enter the `fdisk` command mode. To enter the `fdisk` command mode, type `fdisk` and the disk drive of interest. The `fdisk` command mode screen is displayed and

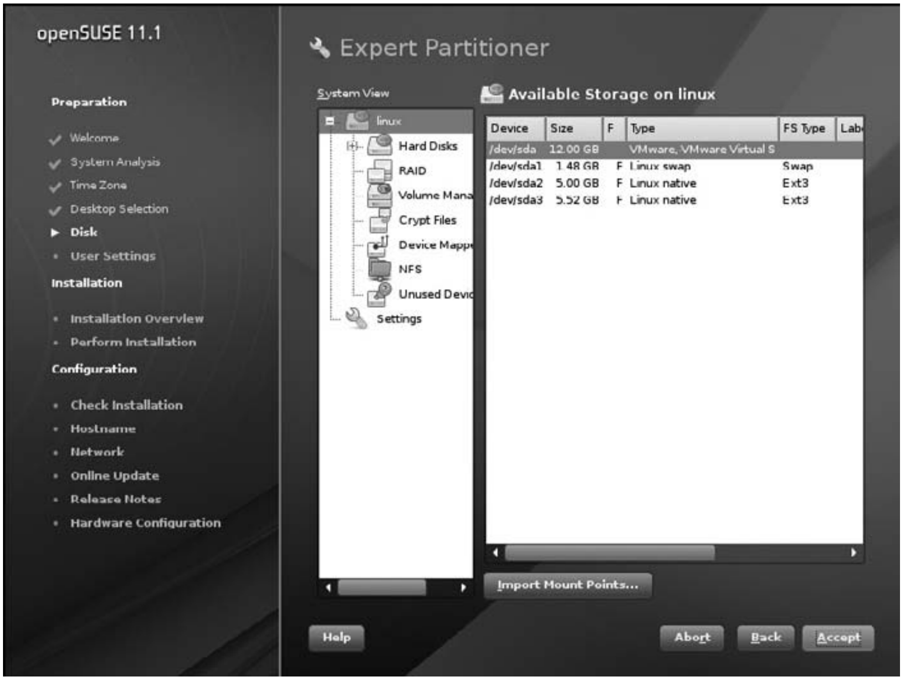


FIGURE 2.24
Expert partitioner.

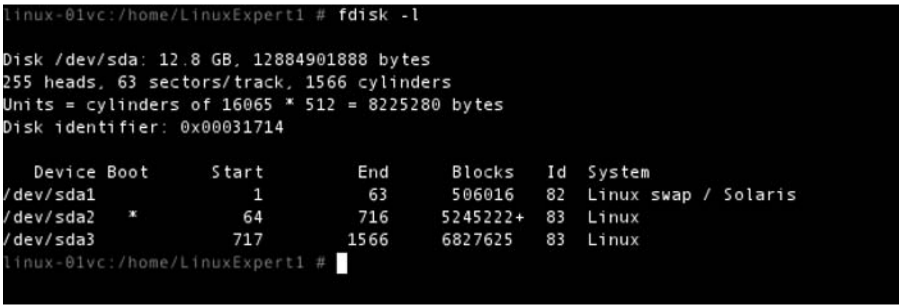


FIGURE 2.25
`fdisk -l` *command.*

you can enter the menu command "`m`" to display a help screen that lists the commands available for use, as shown in Figure 2.26.

The partitions created, whether primary or extended (with logical subdivided partitions), must be assigned a partition type. The partition type is used for hosting specific filesystems. Figure 2.27 presents a listing of the partition types Linux support. The Linux partition type 83 is used to support Linux filesystems. The Linux partition type swap is used to support the swap partition (type 82).

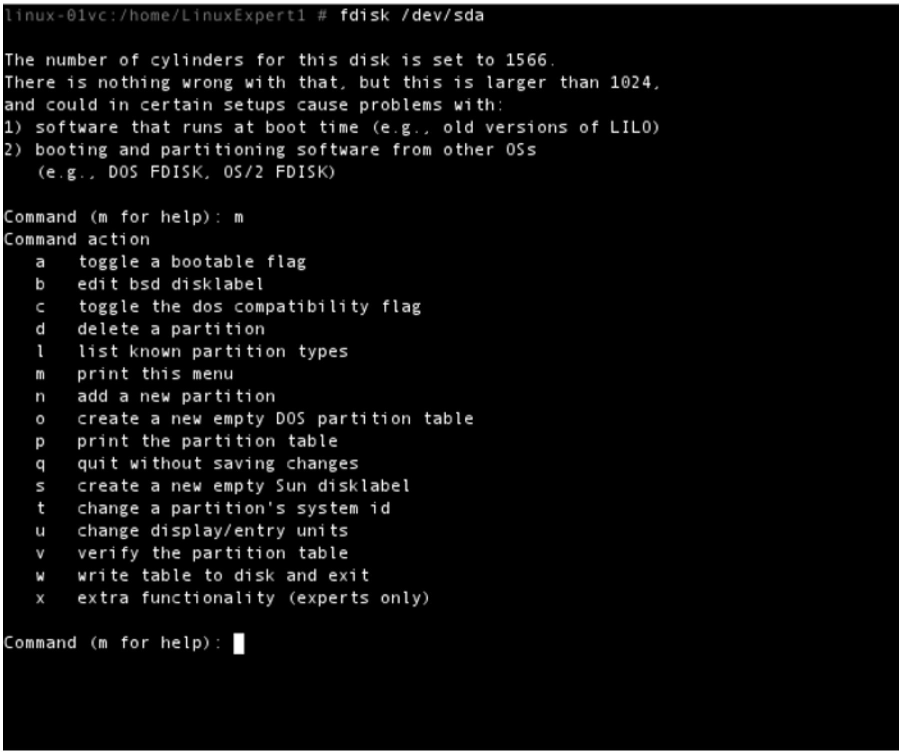


FIGURE 2.26
fdisk help *command.*

To select and configure the Linux partition type, enter the `fdisk -t` command and the number of the partition you want to modify. The `fdisk` command program will prompt you to enter the new type of partition that you want to change the system to. The `fdisk -l` command will display the various partition types supported by openSUSE.

User, application, and system files and folders must be stored on the physical disk. To accomplish this objective, you must implement a filesystem structure on the physical disk. The Linux environment provides support to many different filesystems. Each filesystem has its own unique way of storing files and folders internally for quick access and indexing. Filesystems created normally during the partition creation process will immediately make the newly allocated space available on the hard drive.

Many different filesystems exist for the Linux operating system (for example, ext2, ext3, ReiserFS, JFS, XFS, VFAT/NTFS). Each filesystem offers advantages and disadvantages. One such advantage that can also be a disadvantage is journaling. For some environments, journaling is a critical feature, and filesystems that provide journaling capabilities surface to the top.

```
linux-01vc:/home/LinuxExpert1 # fdisk /dev/sda

The number of cylinders for this disk is set to 1566.
There is nothing wrong with that, but this is larger than 1024,
and could in certain setups cause problems with:
1) software that runs at boot time (e.g., old versions of LILO)
2) booting and partitioning software from other OSs
   (e.g., DOS FDISK, OS/2 FDISK)

Command (m for help): l

 0  Empty           1e  Hidden W95 FAT1 80  Old Minix       bf  Solaris
 1  FAT12           24  NEC DOS         81  Minix / old Lin c1  DRDOS/sec (FAT-
 2  XENIX root      39  Plan 9          82  Linux swap / So c4  DRDOS/sec (FAT-
 3  XENIX usr       3c  PartitionMagic  83  Linux           c6  DRDOS/sec (FAT-
 4  FAT16 <32M      40  Venix 80286     84  OS/2 hidden C:  c7  Syrinx
 5  Extended        41  PPC PReP Boot   85  Linux extended  da  Non-FS data
 6  FAT16           42  SFS             86  NTFS volume set db  CP/M / CTOS / .
 7  HPFS/NTFS       4d  QNX4.x          87  NTFS volume set de  Dell Utility
 8  AIX             4e  QNX4.x 2nd part 88  Linux plaintext df  BootIt
 9  AIX bootable    4f  QNX4.x 3rd part 8e  Linux LVM       e1  DOS access
 a  OS/2 Boot Manag 50  OnTrack DM      93  Amoeba          e3  DOS R/O
 b  W95 FAT32       51  OnTrack DM6 Aux 94  Amoeba BBT      e4  SpeedStor
 c  W95 FAT32 (LBA) 52  CP/M            9f  BSD/OS          eb  BeOS fs
 e  W95 FAT16 (LBA) 53  OnTrack DM6 Aux a0  IBM Thinkpad hi ee  GPT
 f  W95 Ext'd (LBA) 54  OnTrackDM6      a5  FreeBSD         ef  EFI (FAT-12/16/
10  OPUS            55  EZ-Drive        a6  OpenBSD         f0  Linux/PA-RISC b
11  Hidden FAT12    56  Golden Bow      a7  NeXTSTEP        f1  SpeedStor
12  Compaq diagnost 5c  Priam Edisk     a8  Darwin UFS      f4  SpeedStor
14  Hidden FAT16 <3 61  SpeedStor       a9  NetBSD          f2  DOS secondary
16  Hidden FAT16    63  GNU HURD or Sys ab  Darwin boot     fb  VMware VMFS
17  Hidden HPFS/NTF 64  Novell Netware  b7  BSDI fs         fc  VMware VMKCORE
18  AST SmartSleep  65  Novell Netware  b8  BSDI swap       fd  Linux raid auto
1b  Hidden W95 FAT3 70  DiskSecure Mult bb  Boot Wizard hid fe  LANstep
1c  Hidden W95 FAT3 75  PC/IX           be  Solaris boot    ff  BBT

Command (m for help):
```

FIGURE 2.27
Linux partition types supported.

Journaling is a feature implemented in some filesystems to provide a mechanism to temporarily store information in a log (a journal). The changes are stored in a log prior to the changes being implemented within the filesystem. This approach reduces the amount of time required by a system recovering from a crash if the data was not updated to the filesystem. Although journaling is good for overall filesystems, many applications provide their own form of internal logging (journaling) and do not need this functionality built within the filesystem. A major disadvantage of journaling is the impact on system resources (for example, RAM and Disk I/O functions).

There are three very common types of filesystems

- ext2 – The second extended filesystem (ext2), one of the oldest and most popular Linux filesystems, is the industry standard. It is a very reliable filesystem. The lack of journaling support is ext2 greatest weakness.
- ext3 – The third extended filesystem (ext3) expanded the ext2 filesystem. ext3 provides journaling support. It is the default filesystem on many newer versions of Linux.
- ReiserFS – This performs faster than the ext3 or ext2 and supports a larger maximum file structure (8TB).

> Presented in the "Installing from Local Media" section, the openSUSE Installer tool automatically configured the root and home partitions to support the ext3 filesystems. ext3 is the default filesystem for openSUSE 11.1. In addition, the openSUSE Installer configured the swap partition to support the swap filesystem.

Another approach in assigning filesystems to partitions is via the use of the `mkfs` command-line tool. This command-line tool assigns the filesystems (for example, ext2, ext3, and ReiserFS) to partitions. To assign the filesystem, `mkfs` command appends the `-t` option followed by the desired filesystem type, for example

```
mkfs -t ext /dev/sda
```

The assigning of partitions and filesystems to your hard drive may result in the filesystem or partitioning being corrupted or you many need to reclaim disk space from an empty and unmounted partition. To accomplish this task, you can use the `parted` command-line tool. Parted will not only assist you in reclaiming disk space, but it will also copy a filesystem from one partition to the next. During the partition creation process, parted will create the filesystem.

To use the feature, first boot your system from a Linux distribution medium and select **Rescue System** from the options presented, as shown in Figure 2.28. The system will allow you to log in as root (no password is required). To enter the `parted` command mode, type **parted** and the disk drive of interest. For example, `parted /dev/sda`. From the `parted` prompt, you can type **help** or **?**. Figure 2.29 displays the `parted` help command response.

EXERCISE 2.4: Using Rescue System to Access Parted

In this exercise, we will use the **Rescue System** to access the `parted` command and review the help features.

Complete the following:

1. From a Linux bootable Installation Source Media, boot the target machine.
2. This will start the openSUSE Installer. Select **Rescue System**.
3. The system will display "Rescue login:" Enter **root**.
4. No password is required. This can produce a security concern if no physical security is implemented for the target system. The system will display the "Rescue:~#" command.

FIGURE 2.28
Rescue System.

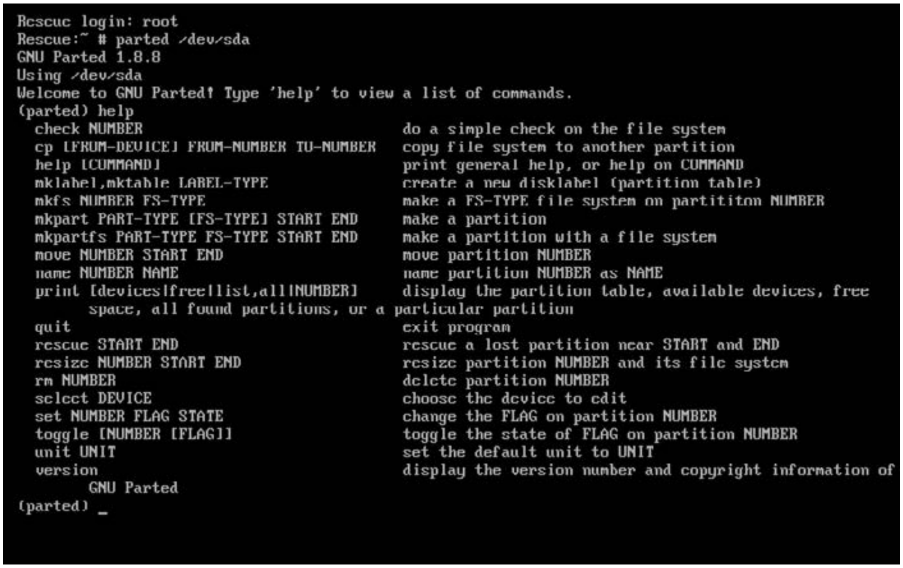


FIGURE 2.29
`parted` *help command.*

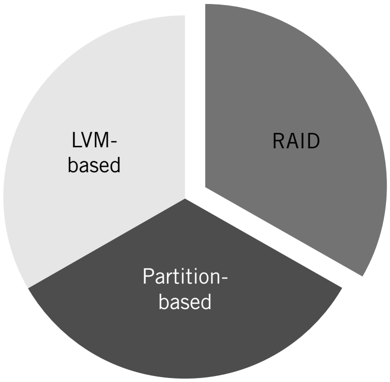


FIGURE 2.30 *Disk types.*

5. At the prompt enter: **parted**.
6. To get help, type **help**. ■

DISK TYPES

In the "Laying Out the Filesystem" section, we discussed various disk drive components required during the installation of the Linux operating system. For example, we reviewed primary and extended partitioning and the different types of filesystems (for example, ext3, ext2, and ReiserFS). For this section, two new disk types will be introduced. The first new type added is the LVM. It is used to create logical volumes. The second new type is the RAID. This type is used to improve performance and/or fault tolerance of the disk subsystem. This brings the total number of disk types to three, as shown in Figure 2.30. The LVM-based and partition-based implementations can be configured on top of RAID or non-RAID systems. The Linux system can also support a hybrid system using all the displayed combinations.

Logical Volume Manager

LVM-based installations offer a unique approach for creating virtual partitions (also known as *logical volumes*). The partition-based approach presented, after implementation, is hard to change. The LVM approach offers greater control of the disk drive environment because you can create virtual partitions that can group physical partitions or disk drives together as one. The command-line tools for LVM are as follows:

- `pvcreate` is a command-line tool for preparing physical volumes for use in LVM.

- `vgcreate` is a command-line tool for creating and naming volume groups.
- `lvcreate` is a command-line tool for creating and naming logical volumes used by filesystems.

EXERCISE 2.5: Using Rescue System to execute `pvcreate`

In this exercise, we will use the **Rescue System** to execute the `pvcreate` command and review the help features.

Complete the following:

1. From a Linux bootable Installation Source Media, boot the target machine.
2. This will start the openSUSE Installer. Select **Rescue System**.
3. The system will display "Rescue login:" Enter **root**.
4. No password is required. This can produce a security concern if no physical security is implemented for the target system. The system will display the "Rescue:~#" prompt.
5. At the prompt enter: `pvcreate -help`. ■

Redundant Array of Independent Disk

RAID is a form of technology available to Linux systems that uses your disk subsystem to provide enhanced read/write performance, protection against data lost due to disk failures, or both. It can be implemented using hardware specific RAID controllers (known as *Hardware RAID*) or RAID functionality embedded within the operating system (known as *Software RAID*). Regardless of whether you use hardware- or software-based RAID, your disk subsystem must include two or more hard disks that will be grouped together into an array to form a virtual hard disk. The advantages and disadvantages are presented below:

- Hardware-based RAID performs faster than the software-based RAID implementation. This is because Software RAID requires more CPU time and has additional memory requirements than Hardware RAID.
- Software-based RAID is operating system dependent and hardware-based RAID is vendor independent.
- Hardware-based RAID is more expensive than software-based RAID. Hardware-based RAID requires you to purchase additional hardware components (for example, RAID controllers).

To implement RAID technology, you must first have an understanding of three basic RAID concepts. The three concepts are as follows: striping, mirroring, and parity. Striping joins the hard disk drives together to form one large disk drive. For example, three 300 MB drives joined together in a striping array will form a single 900 MB drive. Striping evenly writes data across all the disks contained in the array. In addition, striping will evenly read data from all disks contained in the array. This increases your overall disk subsystem performance. The downside to striping is that it does not provide any fault tolerance support.

Mirroring joins the hard disk drives together; however, it does not form one large disk drive. Mirroring forms one disk drive whose size is determined based upon the size of the smallest drive. Mirror writes the same data to both the drives. This approach provides you with a level of redundancy in the event one drive crashes. The disadvantage of mirroring is the impact of having to record the same data twice across two different drives. This reduces the disk subsystem performance. Parity stores information in the disk array subsystem that can be used to rebuild files or lost data in the event one of the disks in the disk subsystem array fails. Unlike striping and mirroring, parity requires a minimum of three disks inside the disk array subsystem.

RAID Levels

To implement striping, mirroring, and parity concepts within your environment, RAID levels are used to describe the different approaches used throughout the industry. The common RAID approaches are presented in Table 2.2. Each RAID level offers advantages and disadvantages, and the best level depends on your requirements. Generally, the partitions should be stored on different drives to get the performance and fault tolerance you want. Do you need a greater performance? Do you need a fault tolerance? If you need both, then RAID 5 is the best solution. In addition to the common levels of RAID, RAID levels can be concatenated (also known as *nested*). This means that the common RAID level numbers are combined with other common RAID levels, sometimes with a "+" in between. For example, RAID 10 (or RAID 1 + 0) consists of a RAID level 1 disk array subsystem, each of which is one of the "drives" of a level 0 disk array subsystem.

During the installation stage, you can elect to configure Hardware RAID or Software RAID subsystems. Hardware RAID installation and configuration are performed in accordance with vendor provided procedures. For Software RAID installation and configuration, the openSUSE expert partitioner tool can be used, as shown in Figure 2.31. The expert partitioner tool allows you to create, edit, and delete Software RAID partitions.

Table 2.2 RAID Levels

RAID Level	Striping	Mirroring	Parity	Description
0	✓			Strips data across two or more hard disk drives within a disk array subsystem. It does not provide data fault tolerance.
1		✓		Mirrors data across two or more hard disk drives within a disk array subsystem. It does provide data fault tolerance.
2	✓		ECC	Provides bit-level striping across five or more hard disk drives within a disk array subsystem. An additional dedicated hard disk drive is used for *Hamming* Code (a form of error correction code) to calculate redundant bits. This level requires a minimum of six disks.
3	✓		✓	Provides byte-level striping data across two or more hard disk drives within a disk array subsystem. An additional dedicated hard disk drive is used for parity. This level requires a minimum of three disks.
4	✓		✓	Provides block-level striping data across two or more hard disk drives within a disk array subsystem. An additional dedicated hard disk drive is used for parity. This level requires a minimum of three disks.
5	✓		✓	Strips data across three or more hard disk drives within a disk array subsystem. A technique known as *distributed parity* is used to also strip the parity bit across each of the disk drives.
6	✓		✓	Strips data across four or more hard disk drives within a disk array subsystem. A technique known as *dual distributed parity* is used to also strip the parity bits across two disk drives.
		Nested RAID Levels		
0+1	✓	✓		This is a striped array set in a mirrored disk array subsystem. A minimum of four disks is required. The number of disks must be even.
1+0	✓	✓		This is a mirrored array set in a striped disk array subsystem. A minimum of four disks is required. The number of disks must be even.
5+0	✓		✓	Data is striped between two or more distributed parity RAID disk subsystems.
5+1	(✓)	✓	✓	Data is mirrored between two or more distributed parity RAID disk subsystems. The (✓) indicates that the striping is performed as a part of the RAID 5 level.

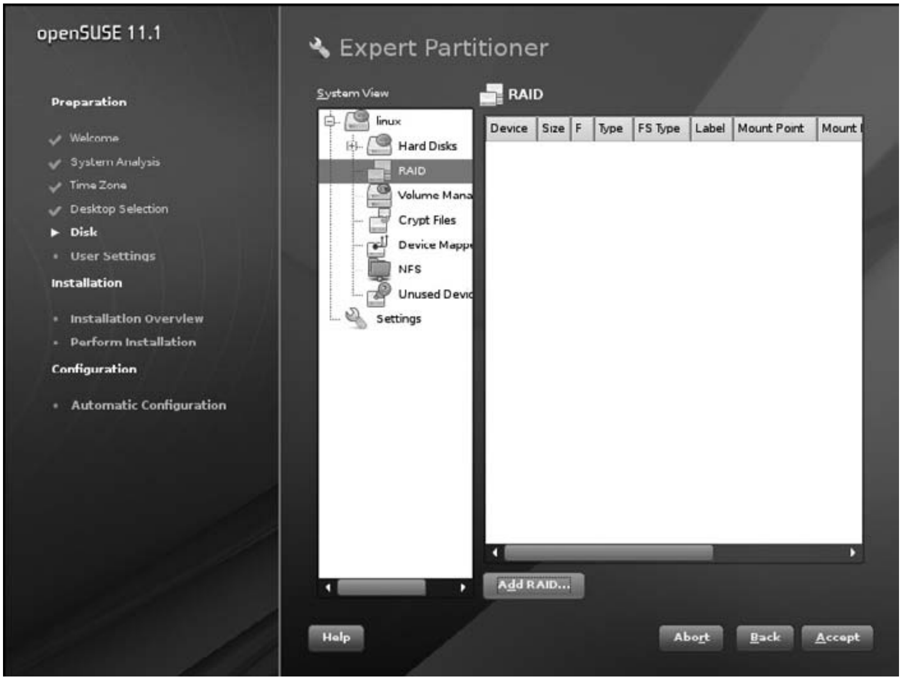


FIGURE 2.31 *Software RAID implementation.*

Software RAID supports RAID Levels 0, 1, and 5. To implement RAID Levels 1 or 0, a minimum of two partitions are required. To implement RAID Level 5, a minimum of three partitions are required. The option **Add RAID** should be selected to display the screen to determine the RAID level and which partitions should be used, as shown in Figure 2.32.

SUMMARY OF EXAM OBJECTIVES

In this chapter, we discussed the Linux distribution installation process and the information you will be required to provide before a successful implementation. For the first section, "A Note about Hardware," an overview of the Linux Hardware Architecture was presented. The Linux Hardware Architecture was divided into four categories. The first category, CPU Architectures, discussed the support CPUs by the Linux environment. This included AMD, Alpha, ARM, IA-64, m68k, MIPS, PA-RISC, PowerPC (for example, Apple, IBM), S/390, SPARC, 32-bit PC-x86-based (for example, AMD, Intel), and 64-bit PC-x86-64 (for example, AMD, Intel). The second category, HAL, presented how a unique tier is layered between the physical hardware and the

FIGURE 2.32 *RAID levels and partition selection screen.*

operating system. The third category, Linux kernel, presented how the kernel functions dynamically by loading modules and instructions for various operating system services. The final category, hardware components, introduced the Linux Hardware Compatibility List, which contains supported hardware devices (for example, printers, monitors, and network cards) for the Linux operating system. Although the Linux+ exam will not ask any specific or complexity hardware installation questions, you should have some general knowledge of the computer systems, the computer peripherals, and the *Linux Hardware Architecture* model.

Next, Installing Linux from Local Media was presented. This section provided in detail the various decisions made to ensure a successful Linux installation from local media (for example, CD, DVD, and .iso image). This process was divided into three stages. The three stages presented were preparation, installation, and configuration. The preparation stage commenced with the various environmental decision you must make. For example, Language, Keyboard type, Clock and Time Zone, User Interface (for example, GNOME, KDE), disk-drive partitioning, and User and Administrator account settings. The next stage, installation, presents you with the information you

provided before performing the actual installation of the Linux operating system. If you agree with the settings, the system will commence installing the Linux operating system. After completing the installation stage, the system initializes the configuration stage. During this stage, the system will either assign default settings (for example, Hostname, Domain Name, Network Configurations, and the installation and configuration of other hardware peripherals) or allow you to manually configure the system. Once the three stages are complete, you can log into the system.

The "Installing across the Network" section presented three different approaches for accessing and installing a Linux distribution across a local or remote network. The three different approaches were based on the HTTP, the FTP, and the NFS protocols. Each approach required network access to a remote network server that stored the Linux distribution within a directory. In each case, you needed to provide the remote server IP address or domain name and the specific directory in which the Linux distribution source resides. For the FTP network installation method, you may need to provide also a user name and password to access the system. For some FTP servers, you may be allowed to connect to the FTP service via the anonymous login.

The "Layout of the Filesystem" section provided a description of the various filesystem types (for example, etx2, ext3, ReiserFS, and NTFS/VFAT) and their advantages and disadvantages. The ext3 filesystem is the default filesystem for most of the modern Linux distributions. The ext3 filesystem supports journaling. The section also discussed hard disk partitioning. Two partition options were presented. The two options were the primary partitions and the extended partitions. Primary partition divides the hard disk drive into physical groups. For most PC-based systems, a maximum of four primary (physical) partitions can be implemented on a hard drive. Next, extended partitions were discussed, and it is used to further subdivide a partition into smaller logical partitions (groups). For any bootable operating system, you must have at least one primary partition. This initial partition will be used by the operating system to store your operating system files. The remaining three partitions can all be the primary partitions, the extended partitions, or a combination of both. To assist with the viewing, editing, and creating partitions, the `fdisk` command was introduced. To assign a filesystem, after creating a partition, the `mkfs` command was introduced. Finally, in this section, the `parted` command was presented. This command is used to reclaim disk space from an empty and unmounted partitions. This may be required if the filesystem or partitioning becomes corrupted or you many need to reclaim disk space from an empty and unmounted partition.

The final section, "Disk Types," presented the LVM and RAID disk types. The LVM feature within Linux is used to create logical volumes. This feature was developed to overcome the fixed partition limitation by not implementing the partition-based approach. The RAID feature was used to improve performance and/or fault tolerance of the disk subsystem. For the RAID feature, various RAID levels were introduced. This included the three critical RAID levels 0, 1, and 5 implemented within a Software RAID environment. To implement RAID technology, three different RAID concepts were presented. The three concepts are striping, mirroring, and parity. Striping joins the hard disk drives together to form one large disk drive. Mirroring joins the hard disk drives together; however, it does not form one large disk drive. Mirroring forms one disk drive whose size is determined based upon the size of the smallest drive. Striping and mirroring can be implemented with at least two disks. Parity stores information in the disk array subsystem that can be used to rebuild files or lost data in the event one of the disks in the disk subsystem array fails. Unlike striping and mirroring, parity requires a minimum of three disks inside the disk array subsystem.

SELF TEST

1. Your manager has asked you to order the next set of workstations for the department. In addition, the organization has decided to migrate from a Microsoft Windows XP operating system to a Linux operating system environment. As a result, the workstations you order must support a Linux operating system. To verify that the workstation you plan on ordering is supported by the Linux distribution you would like to install, what should you do?

- **A.** Configure the workstation to dual boot both Windows 98 and Linux operating systems.
- **B.** Tell your manager, Linux is an operating system for servers only.
- **C.** Review the HCL for the Linux distribution you would like to install to verify the version of Linux you plan on installing supports the workstations you want to procure.
- **D.** Check the Microsoft Web site for additional information about installing Windows XP.

2. Your organization needs a Linux filesystem that supports journaling. Which filesystem supports journaling?

- **A.** ext for VFAT

B. ext2
C. ext3
D. ext5

3. Your organization has decided to implement RAID 5. What is the minimum number of hard disk drives required to support RAID 5?
   A. Zero disk drives are required. RAID 5 does not exist.
   B. Two disk drives are required.
   C. Three disk drives are required.
   D. One disk drive and a Tape Backup system are required.

4. Which protocol does not support the installation of Linux across a network?
   A. HTTP
   B. NFS
   C. FTP
   D. USB

5. When installing a Linux distribution source across a network, which network protocol should you use for anonymous login support?
   A. SMTP
   B. FTP
   C. TELNET
   D. LDAP

6. Which graphical user interface is supported by the Linux operating system?
   A. KDDE
   B. GNOOME
   C. KDE
   D. GMONE

7. What is the maximum number of primary partitions supported on a hard disk drive for a PC-based system?
   A. Five primary partitions are supported.
   B. A hard disk drive cannot support primary partitions.
   C. Four primary partitions are supported.
   D. Only secondary partitions are supported.

8. To perform an HTTP-based network installation, you must enter the following information to establish connectivity with a remote network server.

A. Your workstation IP address and e-mail address.

B. The remote network server's IP address and e-mail address.

C. The remote network server's IP address and remote network server directory containing the Linux distribution source.

D. The remote network server's IP address and your local workstation's directory containing the Linux distribution source.

9. Your organization's management team has decided to implement virtual partition technology. What is the name of the technology within a Linux operating system that supports virtual partitions?

A. Virtual file transfer (VTP)

B. Logical virtual management

C. Disk mirroring system (DMS)

D. Logical volume management (LVM)

10. What are the extended partitions used for on hard disk drives?

A. To further divide a hard disk drive into smaller partitions.

B. Extended partitions are not supported on hard disk drives.

C. Linux does not support extended hard disk drives.

D. Primary partitions and extended partitions cannot coexist on the same hard disk drive.

11. When using the `mkfs` command, what is the `-t` option used for when inserted as a parameter?

A. The `-t` option is used to test the network bandwidth.

B. The `-t` option is used to terminate the operating system.

C. The `-t` option is used to assign filesystems to partitions.

D. There is no `-t` parameter associated with the `mkfs` command.

12. To see all the current disk drives on your system and the current disk geometry, what command should you enter?

A. `mkfs -t`

B. `flpart -l`

C. `fdisk -l`

D. `diskgeo -t`

13. What is the purpose of the `parted` command?
   - A. To reclaim unused disk space
   - B. To establish disk striping
   - C. To implement RAID 5
   - D. To test system's on-board memory for defects
14. You are installing Linux on your organization's server. This is a new installation. You must partition the hard disk for the new Linux installation. Which is the best hard disk partition architecture for supporting root, swap, and home partitions?
   - A. Primary partition architectures should be used for the root and swap partitions and extended partition architecture should be used for home partition.
   - B. The root, swap, and home partitions should all be extended partitions.
   - C. The root and home partitions should be placed on extended partition architectures and the swap partition should be placed on the primary partition.
   - D. Only swap and home should be placed on the primary partition and the root partition should not be used.
15. During the initial Linux installation process, which application is used to test your system's RAM for an x86-based CPU architecture?
   - A. testmemx86
   - B. memtest86
   - C. memtestx86
   - D. memx86test

SELF TEST QUICK ANSWER KEY

1. C
2. C
3. C
4. D
5. B
6. C

7. C
8. C
9. D
10. A
11. C
12. C
13. A
14. A
15. B

This page intentionally left blank

CHAPTER 3

Managing Filesystems

Exam objectives in this chapter

- Filesystem Types
- Mounting and U(n)mounting Filesystems
- Partitions
- Directories
- Filesystem Management

UNIQUE TERMS AND DEFINITIONS

- **Filesystem** A *filesystem* provides the operating system with a framework (a structure) for the storage, organization, modification, removal, and retrieval of digital information. It is responsible for organizing and maintaining files, folders, metadata, and residual data as containers for storing digital information.
- **Swap** The allocation of physical disk space to function as virtual memory when the amount of physical memory (random access memory [RAM]) is full. If the system needs more memory resources and the physical memory is full, inactive pages in memory are moved to the swap space. When the system swaps out pages of memory to the hard disk drive, the system's RAM is freed up to perform additional functions.

- **Server message block file system (SMBFS)** A mountable SMBFS for Linux that allows Windows- or Linux-based workstations access to directory/file shares on a network-based Linux server.
- **Network file system (NFS)** NFS, as presented in Chapter 2, is a framework designed to allow a user on a client workstation to access remotely files over a network on a network-based server.

INTRODUCTION

The Linux+ certified professional needs to have a good understanding of the creation, administration, and management of the Linux filesystem environment. This chapter presents the strategies for the creation of filesystems, the different types of filesystems, the tools used to create filesystems, and the tools used to administer filesystems.

FILESYSTEM TYPES

The computer architecture is comprised of many critical components. In Chapter 2, we discussed many of those critical components (for example, hardware, disk drive, and partition) during the installation of the Linux distribution. For this section, we will discuss Linux filesystem types in greater detail. A filesystem provides the operating system with a framework (a structure) for the storage, organization, modification, removal, and retrieval of digital information.

To achieve this objective, filesystems are responsible for organizing and maintaining files, folders, metadata, and residual data as containers for storing digital information. Regardless of the selected filesystem type, data containers can be as large as terabytes (TB) in size or as small as a sector. The size of a sector can vary among the different filesystem types; it is the smallest amount of disk space that can be assigned to hold a file. The default size of sector is typically 512 bytes.

Each form of data is described below:

- ***Files*** are sectors of allocated space within a filesystem used for storing digital information, which is available to access or execute. The allocated space can be contiguous or noncontiguous. To identify files, files are assigned a filename, which is identified by a sequence of alphanumeric and/or Unicode character sets. In some cases, while not mandatory, a filename can include a dot followed by a series of alphanumeric and/or Unicode character sets (for example, report.doc,

budget.xls) that function as extensions. There are four different types of files within the Linux environment. The four types of files are regular, links, first-in first-out, and sockets.

- ***Directories*** (also known as *folders*) are sectors of allocated space within a filesystem used to group files. This association is performed and maintained within a File Allocation Table (FAT). To identify directories, directories are assigned a directory name, which is identified by a sequence of alphanumeric and/or Unicode character sets. In addition to having names, directories have structures that can be flat or hierarchical. In case of hierarchical structures, directories can contain subdirectories.

- ***Metadata*** is forms of data used by the operating system to further characterize files and folders. Typically, some form of indexing file is created and used. The indexing information can contain the file size, the file date and time stamp, sector location, and information pertaining to access permission and device type. The type and amount of metadata created and maintained vary among filesystems. In addition, metadata can also include backup copies of files and folders for redundancy purposes. This can even include backup copies of metadata files.

- ***Residual data*** is a form of data remaining within the filesystem after the file, filename, folder, and folder name relationship has been severed. This typically occurs after deleting a file or folder. Residual data can vary in size. It can be as small as a sector (this is known a *slack space*). *Slack space* is any residual data remaining on a disk not currently associated with any particular file. It is the space remaining after the last byte of a file and the first byte of the next sector.

For the various different types of filesystems, the storage containers (also known as *storage media types*) can reside on many different storage devices. The storage devices can house static and dynamic generated data. As a result, different filesystem structures exist for the different types of containers. Not all operating systems provide support for the various storage containers. The Linux operating system supports a vast number of storage media types, as shown in Figure 3.1.

Listed below are descriptions of each of the storage media types:

- The *hard disk storage media* type, the most common, is used for the storing of data on hard disk drives. This type of storage device can be connected directly (known as an *internal disk drive*) or indirectly (known as an *external disk drive*) to a computer system. This form of filesystem storage is also known as *local filesystem storage*. This model

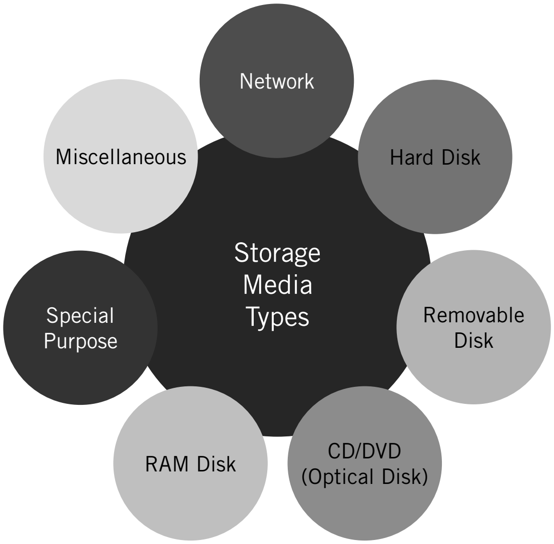


FIGURE 3.1 *Linux storage media types.*

was introduced in Chapter 2; however, greater details will be presented later in this chapter.

- The *optical storage media* type is divided into two different categories. One category supports CDs and the other category supports DVDs. Both CDs and DVDs are supported by the ISO 9660 filesystem standard, also known as the *CD File System* (CDFS), which is used for the storage of data on optical disk media. This includes the support for *Rock Ridge*, *Joliet*, and *El Torito* extensions. The DVD category also supports the Universal Disk Format (UDF). UDF is considered to be a CDFS replacement. UDF is a format specification based on the ISO/IEC 12246 standard.
- The *network-based storage media* is an architectural model comprised of client and server computers. For this model, the client establishes remote network connectivity to a server to access files that reside on the server's hard disk drive. Examples of network-based filesystems include Server Message Block (SMB), Andrew File System (AFS), and the Network File System (NFS). This model was introduced in Chapter 2; however, greater details regarding SMB and NFS will be presented later in this chapter, in the "Filesystem Management" section.
- The *removable storage media* is a filesystem designed for storing files on flash memory devices. Flash memory devices, a form of nonvolatile computer memory, can electrically erase data and save new or resave old data on the device. Flash memory devices are primarily used in memory

cards and Universal Serial Bus (USB) flash drives. Such filesystems include TrueFFS, TFAT, FFS1, JFFS2, and YAFFS.

- The *RAM disk storage media*, which will be presented in Chapter 4, functions as a storage container by using a segment of main memory in RAM. The compressed ROM filesystem (cramfs) is used with many Linux distributions for initrd images (explained in Chapter 4).
- *Special-purpose storage media* are systems implemented to handle files, folders, metadata, and residuals dynamically. Special-purpose filesystems are created by an application or software/system management tool. This includes database applications, registry-based applications, transactional-based systems, and various other unique file-based application systems.

The various Linux storage media types that are accessed can be accessed locally or remotely. Local storage media types, presented in the next section, are storage devices that are directly attached to your client machine. Remote storage media types, presented in the "Network" section, are filesystems accessed by your client machine through a network filesystem.

Local

The local implementation of a filesystem is not uncommon. This approach alleviates many challenges that are typically associated with the implementation of network-based filesystems, RAM disk-based filesystems, and removable filesystems. This includes performance challenges due to limit network bandwidth, access failure due to lost network connectivity, redirection attacks that point your system to a different (perhaps compromised) server, and nonpersistent filesystems (for example, RAM disk) that require reconfiguration once the electrical power is restored.

In Chapter 2, several different local filesystems were introduced during the installation of the Linux system, including the three very common types (for example, ext2, ext3, and ReiserFS). Each filesystem has its own unique way of storing files and folders internally for quick access and indexing. This section will present some additional filesystems that are somewhat common. The filesystems are FAT, New Technology File System (NTFS), and Virtual File Allocation Table (VFAT).

FAT is a widely supported filesystem on most systems, and flash memory devices exist in many different standards. Those standards include FAT12, FAT16, FAT32, and a variation called VFAT. FAT12 has a cluster address limitation of 12-bits. It was designed to support storage devices with a maximum capacity of 16 MB (megabytes). FAT16, which has a cluster address size of 16-bits, was designed to support a maximum partition size of 2 GB

(gigabytes). FAT32, which has a cluster address size of 32-bits, was designed to support a maximum volume size of 2 TB and a maximum file size of 4 GB. The VFAT filesystem introduced before FAT32 was designed to handle long file names. Unlike VFAT, earlier Windows operating systems that used the FAT filesystem allowed files to be named with only eight alphanumeric characters, with a period separating the name from a three-alphanumeric-character extension. This was known as the *8.3 notation*. Windows 95, which introduced the VFAT filesystem, supported long filenames that allowed files to have names up to 255 alphanumeric characters.

Exam Warning

The VFAT filesystem is the preferred filesystem for mounting foreign filesystems.

The NTFS, introduced to support the Microsoft Windows NT operating system, provided several new features unlike the previous filesystems. For example, NTFS provides filesystem security, Unicode, compression, and journaling. The maximum volume size is approximately 256 TB and the maximum file size is 16 EB (Exabytes).

EXERCISE 3.1: Verifying the Current Filesystem Partition Type id

For this exercise, you will enter the Linux `fdisk` command to obtain a listing of the current filesystems partition types implemented and the filesystem partition types available. This exercise displays the specific primary disk drive attached to your system, the corresponding disk geometry, and the associated filesystem.

Complete the following:

1. From the root command prompt, enter the following: `fdisk/dev/sda`.
2. Press the **Enter** (Return) key.
3. From the `fdisk` prompt, enter the `p` option to print the device partition table.
4. Review the system id (for example, 83) for each partition.
5. From the `fdisk` prompt, enter the `l` option to list the known various partition types.
6. Cross-reference your partition type to the list presented.
7. From the `fdisk` prompt, enter the `q` option to quit. ■

Network

In today's intra- and internetworking environments, the Linux filesystem can also span across a network. This allows the Linux filesystem to functions as a client-server model that provides file-sharing services to Linux systems remotely across a network. The server component provides shared directories that can be accessed through network connectivity. The client component, after obtaining access, connects the shared directories to a mount point on the local filesystem. Within the Linux environment, the two primary network-based filesystems are the NFS and the SMB filesystems.

The NFS, as presented in Chapter 2, is a framework designed to allow a user on a client workstation to access remotely files over a network on a network-based server. The NFS is an open standard defined in RFC 1094, RFC 1813, and RFC 3010. In this model, a NFS server shares one or more directories that can be accessed remotely by a network client. The NFS was originally designed to work with the User Datagram Protocol (UDP). Later NFS versions use registered ports TCP 2049 and UDP 2049. Additional details about NFS are presented in the "Filesystem Management" section.

The Server Message Block File System (SMBFS) is a framework designed to allow workstations access to directory/file shares on a network-based server. This model, implemented as client-server architecture, is comprised of two components. The first component, SMB protocol, provides mechanisms for performing interprocess communications. This allows various different types of access (for example, read, write, and delete) to different files on the network-based server and access to other server-side resources (for example, printing). The second component, SMB service, is an application that resides on both the client-side and server-side. The SMB service interoperates with the system's security authentication mechanisms and local filesystem. This allows the SMB environment to interoperate with various other SMB servers (for example, Microsoft Windows, IBM AIX, and Apple MAC OS). The Microsoft version of the SMBFS is known as the *Common Internet File System* (CIFS). There are some subtle differences between the two filesystems; however, interoperability is achieved. For the Linux environment, the Samba application, a free client/server implementation, provides support for both SMBFS and CIFS. Details about the Samba application is presented in Chapter 9. The SMB protocol was originally designed to work with NetBIOS/NetBEUI combination. This approach introduced a large amount of network traffic due to the nature of NetBIOS broadcast. A subsequent Transmission Control Protocol/Internet Protocol version, without NetBIOS overhead, was later released to listen on server port TCP 445.

MOUNTING AND U(N)MOUNTING FILESYSTEMS

The mounting and unmounting of a Linux filesystem is another critical skillset required for the Linux Professional. Unlike the Microsoft Windows operating system, which mounts its filesystems automatically, the Linux operating system allows you to manually mount a filesystem or have a filesystem automatically mounted when your system initially starts up. So, what it mounting? *Mounting* is form of attaching or joining a separate storage device to your existing root directory hierarchy. It makes accessible physically separate disks and/or partitions on a local or remote machine available to you. The attached location on the client machine is called a *mount point*.

The `mount` and `umount` Commands

To manually mount a filesystem to your existing root directory structure, the connection is established to a local mount point, and the Linux professional needs to become very familiar with the Linux `mount` command. The Linux `mount` command can be used with and without any arguments. The Linux `mount` command issued with no arguments (as a stand-alone command) will list all of the currently mounted filesystems on your system, as shown in Figure 3.2.

The Linux `mount` command, when issued with arguments, typically require only two arguments. The first argument represents the storage device that you wish to attach to your root directory structure. A unique feature about Linux operating systems is its treatment of devices (for example, storage devices, terminals). The Linux system treats devices as directories (folders). The /dev directory is used within the Linux environment to mount devices. The second argument represents the directory location where you want to attach it underneath. Again, this is known as the mount point. The

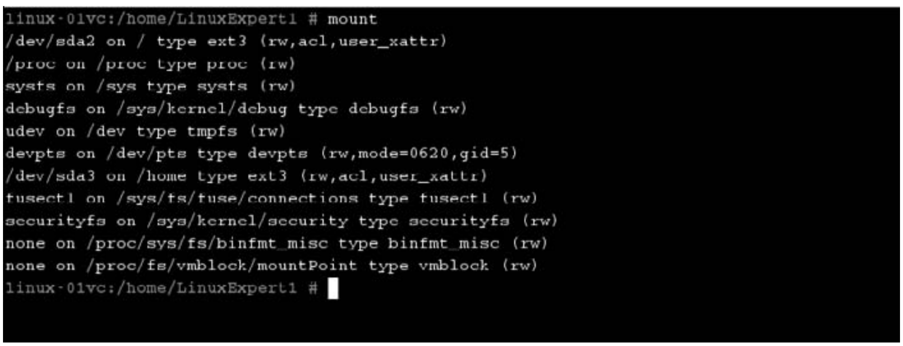


FIGURE 3.2
Linux `mount` *command with no arguments.*

general directory location for most Linux systems is the /mnt and /media directories. The /media directory, typically, is used for the mounting of removable media (for example, floppy disks, CD/DVD drives, and USB/thumb drives). In addition, prior to mounting any storage device, a mount point directory needs to exist before executing the `mount` command.

```
mount/dev/sdb1 /mnt/morestorage
```

The Linux `mount` command includes extra arguments (for example, `sb` for superblock, `noload` for turning off journaling) that can be used, if necessary. To execute the additional arguments, the `-o` argument must be specified first. To specify a specific filesystem type, the `-t` argument should be used. In most cases, the `-t` option is not necessary. The Linux kernel, typically, will detect the type of filesystem of the storage device that you are attaching.

Depending on the type of device you are mounting, at some point of time in the future, you may wish to disconnect the mounted device from the local mount point. To perform this operation, the Linux `umount` command is used. To perform this operation, only one additional argument is required. This argument, where you attached the device, is the mount point.

```
umount/mnt/morestorage
```

> **Note**
> The Linux `umount` command is spelled without the letter (n). Even though its purpose is to unmount the storage device, the actual command omits the letter n.

/etc/fstab

To automatically mount a filesystem, the Linux professional needs to become very familiar with the /etc/fstab file, as shown in Figure 3.3. This text-based file contains the filesystems defined during the installation of the Linux distribution (for example, *root* partition, *swap* partition) and any new filesystems that you would like to mount on a permanent basis whenever the system is booted. Most Linux systems automatically mount removable devices such as CD/DVD, floppy disks, and USB devices.

The structure for adding a filesystem to the /etc/fstab file so that it can be mounted automatically at boot up contains the disk partition you want to mount, the directories mount point, the filesystem type, and other filesystem options (for example, `noload`, `noatime`, and `noauto`).

```
LinuxExpert1@linux-01vc:/etc> cat fstab
/dev/sda1          swap               swap       defaults          0 0
/dev/sda2          /                  ext3       acl,user_xattr    1 1
/dev/sda3          /home              ext3       acl,user_xattr    1 2
proc               /proc              proc       defaults          0 0
sysfs              /sys               sysfs      noauto            0 0
debugfs            /sys/kernel/debug  debugfs    noauto            0 0
usbfs              /proc/bus/usb      usbfs      noauto            0 0
devpts             /dev/pts           devpts     mode=0620,gid=5   0 0
LinuxExpert1@linux-01vc:/etc>
```

FIGURE 3.3
Linux /etc/fstab file.

EXERCISE 3.2: Mounting and Unmounting a Filesystem

For this exercise, you will use the Linux `mount` command and the Linux `umount` command on the disk partition /dev/sdb1. You will attach the disk partition to the mount point /mnt/morestorage (the mount point directory needs to be created in advance using the Linux `mkdir` command).

Complete the following:

1. From the root command prompt, enter the following: `mount/dev/sdb1 /mnt/morestorage`.
2. Press the **Enter** (Return) key.
3. To display the mounted devices enter the following: `mount`.
4. Press the **Enter** (Return) key.
5. To unmount the device, enter the following `umount/mnt/morestorage`.
6. Press the **Enter** (Return) key. ■

PARTITIONS

To implement an effective and efficient Linux storage device infrastructure, the Linux professional must understand the Linux filesystem triad, as shown in Figure 3.4. Partitioning, as implemented in Chapter 2, is the allocation of electronic storage space within a storage device into separate data areas for a specific filesystem type. Whether implemented through a partition editor tool or through the Linux `fdisk` command, partitions can be created, deleted, or modified. The folder and file creation and placement process cannot occur until after the partitioning and formatting of the storage device.

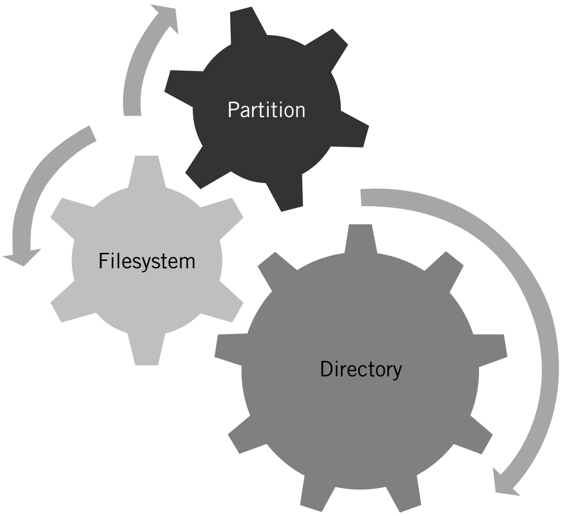


FIGURE 3.4 *Linux filesystem triad.*

The decision to implement one or more partitions per storage device is based on certain advantages and disadvantages. The following are the reasons to create multiple partitions on your storage device:

- Improves access time for data and applications colocated on the same partition.
- Supports the separation of user files and operating system files.
- Provides dedicated operating system swap or system paging file space.
- Protects operating system files from the rapid growth of system data files (for example, logging, system cache) that may consume all available disk space quickly.
- Provides support for multibooting environments.
- Provides a layer of isolation to prevent or protect one partition's system resources from another partition's system resources.
- Allows the implementation of various filesystems and different disk geometry strategies to improve read and/or write performance.

The Linux operating system, like most modern operating systems, supports the creation and use of multiple partitions. After creating a partition, a filesystem (for example, ext2, ext3, and ReiserFS) must be assigned to the filesystem. As presented in Chapter 2, this was implemented through a partition editor tool or through the Linux `mkfs` command. After the filesystem,

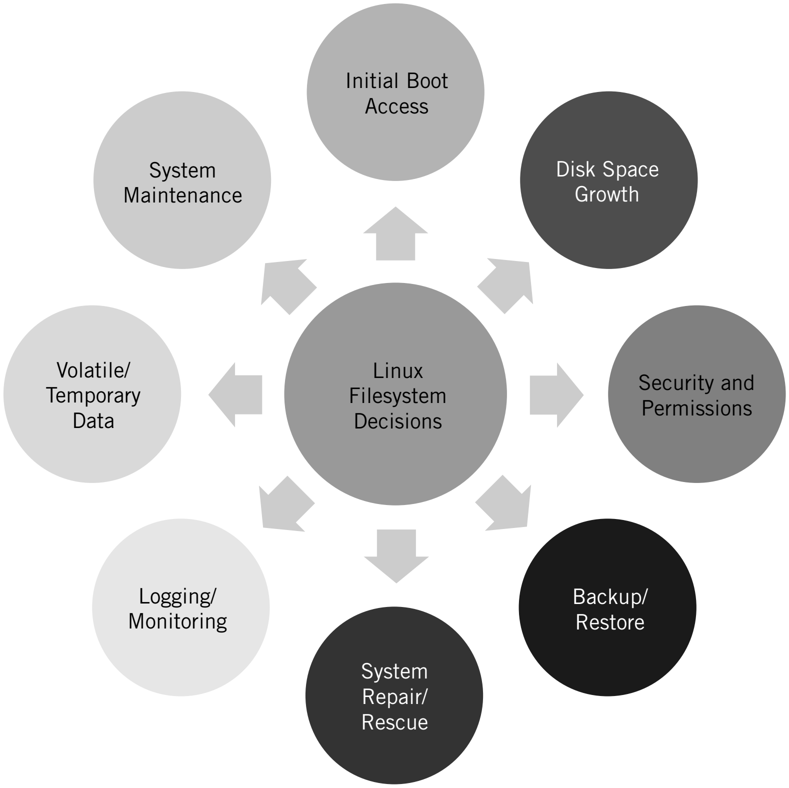


FIGURE 3.5
Linux filesystem decisions.

a directory structure would need to be created for the allocation and organization of files and folders. For the Linux professional, it is the balancing of the Linux filesystem triad that presents the greatest challenge. Each leg of the triad presents a series of tradeoffs. These tradeoffs, as presented in Figure 3.5, are categorized as follows:

- **Initial boot access** During the Linux installation process, it is critical that system BIOS and GRUB functions be able to access a Linux partition. The system BIOS will access the Linux primary partition to execute the bootloader, GRUB. GRUB, which is comprised of two stages, must be able to access the */boot* partition to retrieve the mini Linux kernel and other critical configuration files. As a result, for some installations, placing the initial boot applications and configuration files on a separate Linux partition makes the installation process easier – especially since, during the initial boot process, the mini Linux kernel and *initrd* will have limited access to disk device drivers.

- **Disk space growth** The design and implementation of a Linux system requires disk space growth projections. As a result, the Linux professional must be able to predict the amount of disk space needed for current and future use. A wrong calculation could end up in data not being saved or a corrupted partition. The projections should include installed applications and associated application data and user files. If a fixed partition-based implementation is used, the Linux professional must establish space limitations or quotas. Creating a separate partition for application data and user files is very common. You can install everything of the root partition. However, if an application on the same partition gets corrupted, the impact could damage the entire partition. With a multiple partition implemented, if the corruption occurs within one partition, other partitions will not be corrupted. In addition, failed disk space growth projects for system and user data can also present significant disadvantages for fixed-sized partition Linux systems. The introduction of logical volume management (LVM) reduces disk growth limitations by allowing data in volumes to expand across separate physical disks.
- **Security and permissions** Information stored on the same partition opens the door for various security risks. For example, if the partition is corrupted, all the data could be lost. In addition, directories not properly secured could give hackers read/write/execute access to a directory to load data or fill up a partition with data (for example, illegal videos/music files, or stolen data). Creating separate partitions to separate system resources from user files allows you to place more stringent access controls around system resources.
- **Backup/restore** Creating several partitions for backing up and restoring system resources, applications, application data, and user files provides greater schedule flexibility, reduces backup performance impacts, and can improve restoration times. Separate partitions allow the Linux professional to be more precise in backing and restoring data, thereby minimizing system downtimes.
- **System repair/rescue** Similar in nature to the first bullet listed above, initial boot access, creating a separate partition for critical partitions (including the */boot* partition) would make it easier to repair and/or rescue a corrupted partition.
- **Logging/monitoring** System logging and monitoring is not only a tedious job, but also a never-ending job. As a result, log files must be accessed, reviewed, and in most cases saved for security purposes.

The creation of a separation partition for log files will make it easy to backup, restore, secure, and provide remote access to external systems, if necessary.

- **Volatile/temporary data** Computers and associated applications are constantly creating volatile and temporary data. In most systems, this data now resides on the partition storage device (not just in memory). This created temporary data is a result of the ability of the system and applications to read/write/and execute information located within the temporary directories and swap space. The Linux system automatically creates a swap partition. However, there are other directories (for example, /tmp) that traditionally reside underneath the root directory. These other directories can also outgrow the disk space capacity of the partition.
- **System maintenance** The creation of separate partitions for system administration and maintenance purposes can also make the job easier – for example, the recovery of corrupted partitions and the backup and restoration of data.

For the Linux professional, the decision to create a partition or not to create a partition is a tough job. Clearly, a single-partition system is not a sound system design. However, partitioning of every directory is not feasible, either. Therefore, the Linux professional must find the mean between the two extremes. Table 3.1 provides more insight into the tradeoffs the Linux professional must make.

Table 3.1 Tradeoffs for Specific Linux Directories

Linux Directory	Tradeoff
/swap	The Linux swap partition is used as swap space. It functions as virtual memory. The /swap directory is a predefined partition that has its own predefined filesystem. This directory provides additional memory to your system to run large programs.
/boot	The /boot directory contains files and configuration information needed during the Linux boot process. This includes the mini Linux kernel. The implementation of /boot under a different partition would make system rescues and repairs easier.
/home	The /home directory is normally implemented to host users' account directories and user-specific files. The /home directory should be assigned a separate partition. The reason why this directory should be created on a separate partition is due to multiuser file space growth. In addition, performing backup and restore functions is a lot easier if dynamically changing user files reside on a separate partition.

(*Continued*)

Table 3.1 (*Continued*)

Linux Directory	Tradeoff
/opt	The /opt directory is used to support the installation of add-on application software packages. The directory should reside under the root directory.
/tmp	The /tmp directory is used to support programs that require temporary file space that is volatile. It can reside under the root directory; but if temporary files grow quickly, then your root directory may run out of disk space. This can easily be the result of system or application core dumps. In addition, from a security perspective, the /tmp directory is readable, writeable, and executable by all users and applications. Therefore, a hacker may be able to upload and execute programs in the /tmp directory. Therefore, assigning it to another partition should be considered as an option.
/usr	The /usr directory is a shareable read-only data directory to allow Filesystem Hierarchy Standard (FHS)-compliant hosts access to the same information. It is the largest repository of data. The /usr directory contains user documentation, binaries, libraries, software header files, and X Window material and libraries.
/var	The /var directory contains system- and application-generated information as a result of spooling, logging, and system temporary files. The /var directory should be placed on a separate partition, if possible. It is normally a subdirectory underneath the root directory. The primary reason why it should be separate is due to the possible growth of the log files and user mailboxes. This approach would reduce the size of the root partition.

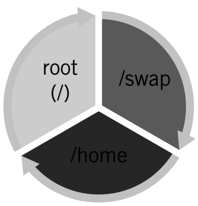


FIGURE 3.6 *Linux filesystem most common partitions layout.*

The recommended partitioning schema, first introduced in Chapter 2, represents the most common approach, as shown in Figure 3.6. This approach entails having two primary partitions and one extended partition. Figure 3.7 presents the same partitions using the Linux `fdisk -l` command. The two primary partitions support the Linux *root* partition and the Linux *swap* partition. The extended partition supports the *home* partition.

EXERCISE 3.3: Adding a New VFAT Partition Type

For this exercise, you will enter the Linux `fdisk` command to add and implement a VFAT filesystem partition type. For the exercise, the secondary Small

FIGURE 3.7
Linux `fdisk` *command displays the most common partitions layout.*

```
linux-01vc:/home/LinuxExpert1 # fdisk -l

Disk /dev/sda: 12.8 GB, 12884901888 bytes
255 heads, 63 sectors/track, 1566 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Disk identifier: 0x00031714

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1               1          63      506016   82  Linux swap / Solaris
/dev/sda2   *          64         716     5245222+  83  Linux
/dev/sda3             717        1566     6827625   83  Linux
linux-01vc:/home/LinuxExpert1 #
```

Computer System Interface (SCSI) disk drive (/dev/sdb) has been added to the system.

Complete the following:

1. From the root command prompt, enter the following: `fdisk/dev/sdb`.
2. Press the **Enter** (Return) key.
3. From the `fdisk` prompt, enter the `l` option to list the known various partition types, and find the Partition Type System id Hex Code for: WIN95 FAT32 (LBA).
4. Press the **Enter** (Return) key.
5. From the `fdisk` prompt, enter the `n` option to add a new partition.
6. Press the **Enter** (Return) key.
7. From within new partition prompt, enter `p` to create a new primary partition.
8. Press the **Enter** (Return) key.
9. Since this is the first/only partition on this device, enter ***1*** as the partition number.
10. Press the **Enter** (Return) key.
11. Select the default for the First Cylinder. Press the **Enter** (Return) key.
12. Select the default for the Last Cylinder. Press the **Enter** (Return) key.
13. From the `fdisk` prompt, enter the `t` option to change the partition type.
14. Enter the Partition Type System id Hex Code WIN95 FAT32 (LBA): `c`
15. Press the **Enter** (Return) key.
16. From the `fdisk` prompt, enter the `p` option to print the device partition table.

17. Press the **Enter** (Return) key.
18. From the `fdisk` prompt, enter the `w` option to write table to disk and exit.
19. Press the **Enter** (Return) key.
20. To create the VFAT filesystem, enter `mkfs -t vfat /dev/sdb1`.
21. Press the **Enter** (Return) key. ■

DIRECTORIES

The Linux filesystem is a hierarchical structure. This structure is used to organize directories (folders) and files. The Linux directory structure is based on the Filesystem Hierarchy Standard (FHS). The current version is 2.3 (made available January 29, 2004). The purpose of the FHS is to provide a reference for members of the Linux community with guidance regarding the use and implementation of the Linux directory structure. In essence, the goal of FHS is to ensure Linux interoperability consistency for users and applications. This will allow users and applications to know the common location of installed files and directories.

Note
For information about the FHS, visit the www.pathname.com/fhs Web site.

The purpose of this section is to provide the Linux professional with an overview of the purpose key Linux directories listed below, as shown in Table 3.2. As a hierarchical tree structure, the filesystem starts at the top with a directory indicated by a forward root. This directory, called the *root* directory, contains all underlying files and directories.

Exam Warning

Linux files and folders (directory names) are case-sensitive.

When addressing all other second-tier directories and files, use the full-path naming convention; all subsequent files and folders use the forward slash to indicate their hierarchical position, to prevent confusion with other directories that could have the same name but located at a different sublayer tiers (for example, third level, fourth level).

Table 3.2 Critical Linux Directories

Linux Directory	Purpose
/	Pronounced "root," the root directory is the top-tier directory. It is the most important directory and is required to boot the Linux system. It contains core directories and files. This includes utilities, configuration files, bootloader information, and start-up information required for the initialization of the Linux system.
/bin	The /bin directory contains Linux commands used by the system administrator and users. The commands in this directory are also accessible when the system is in single-user mode.
/dev	A unique feature about Linux operating systems is its treatment of devices (for example, terminals). The Linux system treats devices as directories (folders). The /dev directory is used within the Linux environment to mount devices.
/etc	To control the execution of programs and support the dynamic Linux environment, the /etc directory is used. This directory contains Linux system and application configuration files.
/media	The /media directory is used to mount removable media (for example, floppy disks, CD/DVD drives, and USB/Thumb Drives) for access by the system administrator and users.
/mnt	The /mnt directory, similar to the /media directory, is used to temporarily mount filesystems.
/proc	The /proc directory functions as a virtual filesystem for system processes and the Linux kernel.
/root	The /root directory, not to be confused with the root directory, is the home directory assigned to the *root* user account.
/sbin	The /sbin directory is used to contain Linux utilities that are used only by the Linux system administrator (for example, root). It contains executables for critical booting, restoring, recovering, and/or repairing the Linux system.
/usr/bin	The /usr/bin directory, unlike the /sbin directory, contains the primary Linux executable commands on the system. Linux users and the root user can execute the commands in this directory.
/usr/lib	The /usr/lib directory contains software libraries and packages for programs. This includes object files and internal binaries that are platform-specific.
/usr/lib64	The /usr/lib64 directory performs the same function as the /usr/lib directory, but for an alternative binary format. This directory supports the 64-bit architecture.

(*Continued*)

Table 3.2 (*Continued*)

Linux Directory	Purpose
/usr/local	The /usr/local directory is the location for locally installed applications by the Linux system administrator. Software installed in this directory typically is not affected by system software updates. In addition, software installed in this directory can be shared.
/usr/share	The /usr/share directory is used to store read-only architecture neutral files. The files contained in this directory are platform-independent.
/var/log	The /var/log directory contains data files generated as a result of spooling, logging, and system temporary files.

EXERCISE 3.4: Comparing Linux Filesystem Directories

In this exercise, we will compare the Linux directory structure for your system with the FHS.

Complete the following:

1. Open a Web browser from a workstation and navigate to www.pathname.com/fhs.
2. Download the FHS document.
3. From the root command prompt, navigate to the root directory enter the following: ***cd /***.
4. From the root directory, compare the directories listed in the FHS document to your Linux directories (for example, /bin, /dev, /sbin, mnt, /etc, /var). Do there match? Are there differences? Why? ■

FILESYSTEM MANAGEMENT

A filesystem environment on a daily basis is constantly changing. This section presents the tools needed to manage, locally and remotely, a Linux filesystem. It includes determining the amount of disk space used and remaining, the establishing of disk space limitations for users, the repairing of corrupted filesystems, the mounting of unique loopback filesystems, the accessing of remote filesystems, and the preparation of swap files or partitions.

Learn By Example: The Hard Way!

An online customer contacted me concerning the performance of a Linux server that always crashes intermittently each month. As always, I started the engagement after the signing of the contract to commence my analysis. I reviewed the customer's intrusion-detecting logs, antivirus logs, firewall logs, application logs, logs located in the /var directory, and I even interviewed the Linux administrator.

After careful review, I found the problem. I informed the customer that I had good news and bad news. For the good news, the problem can be resolved quickly. It appears that a certain user within the IT department (the Linux administrator), on new movie release days (mostly on Fridays), downloads a pirated copy of the movie through the Internet. The individual saves the movie file on the server and burns a DVD copy to take home. The server performance degradation occurred because the server's disk partitions and swap space was at capacity. Periodically, to make space the Linux administrator would delete system and user files to increase disk space.

The solution to the problem was to readdress disk space allocation, separate user files and swap space from the root (/) directory, create disk quotas for all users, and conduct routine system-wide filesystem management for all users (especially root users). In addition, conduct periodic independent security assessment reviews of the entire Linux server environment.

Oh! What about the bad news? I informed the customer that the Linux administrator has been ignoring requests from the Recording Industry Association of America (RIAA) for the last 2 years, and the RIAA has already filed a motion with the court to conduct an electronic discovery. The RIAA is group representing the recording industry distributors in the United States and is very concerned about intellectual property (IP) piracy.

Checking Disk Usage

A task that all system administrators perform is constant-file management. This form of management commences with the ability to review disk space used at the partition level, the directory level, and down to the file level. The Linux `du` command provides a summary of disk space used per file in the current directory and disk space allocated for files contained in any subdirectories. For example, when the Linux `du` command is executed from within a user's home directory, the command will display how much disk space that directory is occupying and any subdirectories underneath the user's home directory.

The Linux `df` command provides summary of the amount of disk space available on a filesystem. The Linux `df` command can present disk space availability in many different ways by using different arguments. The `df -i` argument presents display information about inodes rather that file blocks. The `df -h`, see Figure 3.8, presents disk space summary in a easy-to-understand output format using kilobytes, megabytes, and/or gigabytes.

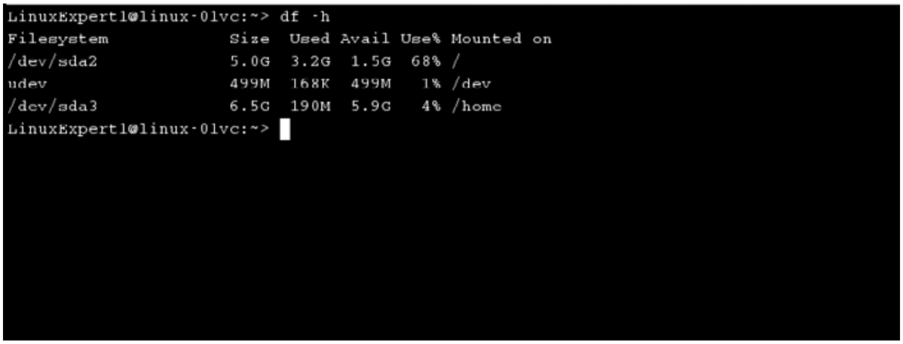


FIGURE 3.8 *Linux* `df -h` *command.*

EXERCISE 3.5: Displaying Disk Summary Information

For this exercise, you will enter three different disk summary commands to obtain disk summary, filesystem type, disk usage, and disk space availability information.

Complete the following:

1. The command mount shows which filesystem (device and type) is mounted at which mount point. From the root command prompt, enter the command: `mount`.
2. Press the **Enter** (Return) key.
3. To display the total size of all the files in a given directory and its subdirectories, the Linux `du` command is used. The parameter `-h` transforms the output into an easily readable format. From the root command prompt, enter the command: `du -h`.
4. Press the **Enter** (Return) key.
5. To obtain information about total usage of the filesystems, the Linux `df` command is used. The parameter `-h` transforms the output into an easily readable format. From the root command prompt, enter the command: `df -h`.
6. Press the **Enter** (Return) key. ■

Quotas

The management of disk space usage for users and applications is a continuous task. Users and errant applications, without limitations, can continue to add data to a folder that could eventually use up all the available disk space on a partition. Once a partition is full of data, not only will users and applications not be able to save information to disk, but also a full partition can also cause system or application damage. To better manage a system, the

implementation of disk quotas can provide valuable filesystem management support. It will allow you to specify limits on disk storage that may be allocated to a user or a group of users. This is accomplished by forcing the users to stay under a prescribed disk consumption limit. This takes away the user's ability to consume unlimited disk space. Disk quotas, implemented across a filesystem, can be configured for individual users and groups.

To use disk quotas for your system, a multistage process must occur. The first stage requires modifications to the /etc/fstab file. To implement disk quotas, you must add the following qualifiers "`usrquota`" or "`grpquota`" to each desired partition in the /etc/fstab file. After modifying the file, the easiest way to activate the disk quota system is to reboot your system. The second stage, the Linux `quotaheck` command, examines the quota-selected filesystems and builds a table of the current disk usage for each filesystem with disk quota enabled. This information is stored in aquota.group and aquota.user files. The final stage is the assigning of disk quotas using the Linux `edquota` command. The Linux `edquota` command, a quota editor, is used to display and change the various quota settings for one or more users or groups.

After the disk quota system is implemented, a summarize disk quota review for a filesystem is available. The Linux `repquota` command is used to create and summarize the disk quota report. The report includes a summary of the disk quotas for the specified filesystems and summaries for the current number of files and amount of disk space per user. Finally, in case of a system crash and other filesystem failures, the Linux `quotacheck` command is used to scan a filesystem for disk quota use and to create, check, and (if necessary) repair disk quota systems.

Check and Repair Filesystems

Linux operating systems, like all other operating systems, have filesystems that will experience some type of disk problem. To maintain the healthiness of your filesystem, you should check, and (if necessary) repair your filesystem. To perform this function within the Linux environment, the Linux *fsck* command is used. This command checks for and repairs Linux filesystem problems. If any problems are identified, the command will display the problem.

Before using the Linux `fsck` command on a filesystem, you must `umount` the filesystem before checking for problems. To use the Linux `fsck` command, you must also provide the name of the filesystem (for example, /dev/sdb1) that will be examined. Presented below is an example using the Linux `fsck` command.

```
fsck /dev/sdb1
```

Loopback Devices

The Linux operating system offers support for an additional unique type of filesystem. This type of filesystem is known as the loopback filesystem. Most Linux distributions have the loopback device compiled into the kernel. The kernel supports the transformation of a special file containing an image of another filesystem into a device that can be used like any other Linux partition or device. Linux loopback devices are commonly used for CD/DVD ISO images. The disk image created of the CD/DVD disc contains the UDF or ISO 9660 filesystem format. Prior to accessing the loopback device, the ISO image must be downloaded and mounted. The Linux `mount` command is used to attach the virtual filesystem image.

EXERCISE 3.6: Mounting an ISO Filesystem

For this exercise, you will mount a ISO image file that contains a mini Linux filesystem.

For this exercise, we will use the following:

- **Server/Domain Name:** http://download.opensuse.org
- **Directory/Folder Location:** distribution/11.1/repo/oss/

  Complete the following:

1. Open a Web browser from a workstation and navigate to http://software.opensuse.org/.
2. Follow the instructions to download the openSUSE 11.1 network installation Boot CD. The downloaded network installation file is an .iso image that contains a bootable Linux distribution. Save the file as MiniCD.iso.
3. From the root command prompt, to create a mount point directory, enter the following command: `mkdir -p /mnt/disk1`.
4. Press the **Enter** (Return) key.
5. Enter the next command to mount the MiniCD.iso file. `mount -o loop MiniCD1.iso /mnt/disk1`.
6. Press the **Enter** (Return) key.
7. Enter the command to change to the new filesystem contained inside the MiniCD.iso file: `cd /mnt/disk1`.
8. Press the **Enter** (Return) key.
9. Type the following command to list the contents of the filesystem: `ls -l`.
10. Press the **Enter** (Return) key. ■

Network File System

Mentioned earlier in this chapter, see "Filesystem Types," NFS is a client/server model designed to make specified directories on a server available to a select subset of clients or all clients in a network. Prior to accessing directories for use, the server-side and client-side must be configured properly. The server-side must configure directories for sharing. These directories are also known as *exports*. The client-side must be configured for mounting the exports.

Commencing with the server-side assumption that the kernel-based NFS service is available – if not, the service can be set to run automatically by including the NFS service in the server's default runlevel – the shares made available are listed in the /etc/exports file, as shown in Figure 3.9. This file contains a listing of directories (exports) and the client machines that may mount the exports. Each line represents a shared directory and any associated options (for example, permissions).

The parameters for each line are as follows:

```
exported_directory <client1> (<options>) <clientN> (<options>)
```

- The first parameter, exported_directory, represents the directory being exported on the server.
- The second parameter, <client1>, presents the host or network to which the export is being shared for access. The client1 parameter can be a single host based on IP address, hostname, or domain name; a wildcard (*) for a group of machines; or an IP network range.
- The third parameter, <options>, represents the options imposed on the connection. The options include read-only (ro), read-write

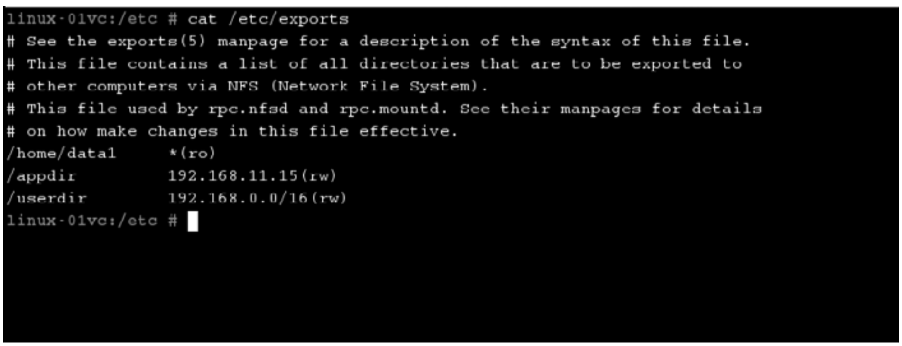


FIGURE 3.9
NFS /etc/exports file.

(rw), `root_squash` (to prevent remote root access), `no_root_squashing` (to allow remote root access), and others.

Below are examples of lines inserted in the /etc/exports file, as shown in Figure 3.9:

- **/home/data1 *(ro)** This exports the /home/data1 directory to all clients with read-only permission.
- **/appdir 192.168.11.15 (rw)** This exports the /appdir directory to the specific client with IP address 192.168.11.15 with read-write permission.
- **/userdir 192.168.0.0/16 (rw)** This exports the /userdir directory to all client within the IP subnetwork 192.168.0.0 range with read-write permission.

Note
The directory listing in the /etc/exports should be configured with the most restrictive access possible. This means not using wildcards for host machines, not allowing remote root-level write access to shares, and mounting read-only shares wherever possible.

To activate the access of shared directories (exports), the Linux *exportfs* command can be used. The following set of arguments can be used with the command:

`-a` This option loads and exports all directories listed in the /etc/exports file.

`-r` This option rereads the /etc/exports file after changes have been made to the share permissions.

`-i` This option ignores the /etc/exports file and exports a directory not listed in the file.

`-u` This option removes (unexport) currently listed exported directories.

`-au` This option removes all currently exported directories.

To determine information about the shared directories (exports) on a server, the Linux `showmount` command can be used. The command will list any exports currently shared, including those listed in the /etc/exports file (if they are currently being shared on the server), as shown in Figure 3.10.

```
linux-01vc:/etc #  showmount -d  server1
Directories on server1:
/home/data1
/appdir
/userdir
linux-01vc:/etc #
```

FIGURE 3.10
NFS `showmount -d` *command.*

Table 3.3 Linux `showmount` Options

`Showmount` **Option**	**Purpose**
`-a`, -*all*	This option, using the format *hostname*:*directory*, where *hostname* is the name of the client and *directory* is the mounted directory.
`-d`, -*directories*	This option lists client-mounted directories.
`-e`, -*exports*	This option prints the servers list of exported filesystems.
`-h`, -*help*	This option provides help summary.

The Linux `showmount` command includes the following arguments:
`showmount [options] [server1]`

The default value for server1 is the value returned by *hostname*. With no options, the command shows the clients that have mounted directories from the host. Some of the available options are shown in Table 3.3.

The client-side, as stated earlier in the section, also requires configuring. First, you must know what NFS servers are available and the associated shares that allow connections. To establish a connection to the share, the client can use the Linux `mount` command. The following demonstrates the way a client can establish an NFS connection:

```
mount server1:/share /mount_point
```

This command will allow you to see the files contained underneath the directory /share on the server1 by changing to the /mount_point directory on the client's machine.

If connecting to a NFS share manually each time is something you do not want to do, you can establish an automatic mounted connection to the NFS share at boot time. This can be accomplished by inserting a line entry to the /etc/fstab file.

Swap

The Linux operating system, like most modern operating systems, requires virtual memory to ensure its successful performance. This virtual memory

can exist as a file or as an entire partition for storage. Within the Linux environment, virtual memory is accomplished by dividing the system's physical RAM into units known as pages, and transferring less frequently used physical units of RAM (pages) to the hard disk drive. This process is known as swapping. When the system swaps out pages of memory to the hard disk drive, the system's RAM is freed up to perform additional functions.

While swapping offers advantages to the Linux system by extending its access to more memory, pages stored and retrieved on the hard disk drive are slower than accessing pages that only resided in physical memory (RAM). For the Linux system to use swap space, a special file or swap partition must be created first. Earlier in Chapter 2, the Linux swap partition was created. The creation of a swap file entails the creation of a special file. Once create, the special file must be designated to function as a swap file so that the Linux kernel will know to use it as swap space. Finally, the designate swap file must be activated.

The swap file, a special file, can be created with Linux `dd` command. Below is an example of the creation of empty Linux swap file.

```
dd if=/dev/zero of=/newswapfile bs=1024 count=1048576
```

The command creates a swap file named "newswapfile." The input file "/dev/zero" is a special Linux file that provides null characters. The newly created swap file is 1 GB in size.

To designate a partition or special file to be used as swap space, the Linux `mkswap` command is used. This command sets up a Linux swap area on a partition or special file. For a partition, you will need to prepare it using the `mkswap` command as root, as follows:

```
mkswap /dev/hdb1
```

For a special file, just as you would use the Linux `mkswap` command to create a swap partition, but this time you would use the name of the swap file as follows:

```
mkswap /newswapfile
```

The Linux `swapon` command is used to designate the specific devices or files on which paging and swapping is to take place. For the partition, you would need to prepare it using the `swapon` command as root:

```
swapon /dev/hdb1
```

For a special file, just as you would use the Linux `swapon` command to designate a swap partition, but this time you would use the name of the swap file as follows:

```
swapon/newswapfile
```

To ensure that the designated swap file or swap partition is being used, the `swapon -s` command will display the current status. To disable or turn off the swap files or swap partitions, the Linux `swapoff` command can be used. The Linux `swapoff` command disables swapping on the specified partition or file. When the `-a` flag is used, swapping is disabled on all known swap devices and files. The Linux `swapinfo` command prints information about the swap partition and swap file.

SUMMARY OF EXAM OBJECTIVES

In this chapter, you reviewed the management of Linux filesystems. A filesystem provides the operating system with a framework (a structure) for the storage, organization, modification, removal, and retrieval of digital information. Filesystems are responsible for organizing and maintaining files, folders, metadata, and residual data as containers for storing digital information. Regardless of the selected filesystem type, data containers can be as large as TB in size or as small as a sector. In addition, you learned the different filesystem types that are available for local and network access (for example, SMB and NFS).

Next, you learned about the mounting and unmounting of filesystems. This entailed learning that the Linux final system, unlike other operating systems, allows you to manually mount a filesystem or have it automatically mounted. For the manual mounting of filesystems, you learned about the Linux *mount* command and the Linux `umount` command to unmount a filesystem. Regarding the process for having filesystems automatically mounted, you reviewed the /etc/fstab file and how filesystems are included during Linux system initial boot-up.

After mounting and unmounting filesystems, the chapter provides details about partitions. This included the design strategies, advantages and disadvantages for creating partitions, the typical partitions implemented, the use of the Linux `fdisk` command, and the use of the Linux `fdisk` command. The Linux `fdisk` command is used to create a partition. After creating a partition, a filesystem (for example, ext2, ext3, and ReiserFS) must be assigned to the filesystem. The Linux `mkfs` command assigns filesystem to the partition.

After creating a filesystem, we discussed the Linux filesystem as a hierarchical structure. This structure is used to organize directories (folders) and files. The Linux directory structure is based on the FHS. The purpose of the FHS is to provide a reference for members of the Linux community with guidance regarding the use and implementation of the Linux directory structure.

In essence, the FHS goal is to ensure Linux interoperability consistency for users and applications. This will allow users and applications to know the common location of installed files and directories.

The final section, "Filesystem Management," entailed the tools needed to manage locally and remotely a Linux filesystem. It includes commands to determine the amount of disk space used and remaining free space (for example, `du`, `df`), the various commands for implementing and managing disk space limitations for users (for example, `edquota`, `quotacheck`, `repquota`), the repairing of corrupted filesystems using the Linux `fsck` command, the mounting of unique loopback filesystems, the accessing of remote NFS filesystems, and the preparation of swap files or partitions using the Linux commands `mkswap`, `swapon`, `swapoff`, and `swapinfo`.

SELF TEST

1. Which Linux command is used to assign a filesystem to a partition?

A. `filesys`

B. `mkfs`

C. `fsmake`

D. `grub`

2. The Network File System uses which registered port?

A. TCP 2049

B. TCP 80

C. TCP 23

D. TCP 25

3. What is the purpose of the /root directory?

A. It is the main directory for all files and system partitions.

B. It provides virtual memory space.

C. It functions as the home directory for the root user.

D. It is a shareable read-only directory for all users to access.

4. What is the role of the /home directory?

A. It is the location for temporary file space.

B. It provides virtual memory space.

C. It functions as the home directory for the typical user.

D. It is a shareable read-only directory for all users to access.

5. What does FHS stand for?
   - **A.** Free home space
   - **B.** Similar to NFS, but works on an Apple Mac
   - **C.** File Hierarchy Specification
   - **D.** Filesystem Hierarchy Standard

6. Which Linux command is used to attach a separate storage device to an existing directory?
   - **A.** `mkmount`
   - **B.** `mount`
   - **C.** `umount`
   - **D.** `fdisk`

7. What is contained in the /var/log directory?
   - **A.** A variation in system device drivers
   - **B.** Contains data as the result of spooling, logging, and system temporary files.
   - **C.** A shareable read-only directory for all users to access
   - **D.** System libraries and packages

8. What argument do you use to obtain an easy readable output for the Linux `du` command?
   - **A.** `-h`
   - **B.** `-i`
   - **C.** `-v`
   - **D.** (no options)

9. Your manager has asked you to mount a CD disc on the community workstation in the lobby, so that everyone can access it. The CD disc needs to be mounted on the /media/cdplayer directory. Which `-t` filesystem option must you include?
   - **A.** `-t` iso9660
   - **B.** `-t` iso
   - **C.** `-t` iso9000
   - **D.** `-t` ext3

10. What is another format for DVDs besides the ISO9660 format?
    - **A.** */swap*
    - **B.** SCSI
    - **C.** Universal Disk Format (UDF)
    - **D.** SMBFS

11. You need to use `fdisk` to establish a partition for a new SCSI disk drive you want to add for extra storage space. The original drives all are IDE drives. Which is the correct syntax?

    A. `fdisk /dev/SCSI1`

    B. `fdisk /dev/IDE`

    C. `fdisk /dev/sda`

    D. `fdisk /dev/sdb`

12. Which file, when the system initially starts up, will automatically mount filesystems?

    A. /etc/fstab

    B. /boot/fstab

    C. /dev/devices.map

    D. /etc/grub.conf

13. What is an ISO loopback device?

    A. The transformation of a special file into a virtual Linux filesystem

    B. A device that returns feedback tests to the monitor

    C. The /null driver device

    D. The IP address 127.0.0.1

14. Which Linux command is used to designate a specific file or partition for swapping?

    A. `/swap`

    B. `fileswap`

    C. `swapon`

    D. `grub`

15. What is the purpose of the Linux `exportfs` command?

    A. It functions as the Linux bootloader

    B. To partition a storage device

    C. To designate a specific file or partition for swapping

    D. To activate the access of shared NFS directories

SELF TEST QUICK ANSWER KEY

1. B
2. A
3. C
4. C
5. D
6. B
7. B
8. A
9. A
10. C
11. C
12. A
13. A
14. C
15. D

CHAPTER 4

Booting Linux

Exam objectives in this chapter

- GRUB
- Runlevels
- Troubleshooting Boot Issues

UNIQUE TERMS AND DEFINITIONS

- **GRand Unified Bootloader (GRUB)** An application used on most modern versions of the Linux operating system. It is a dynamically configurable program used to perform a sequence of events on a computer to load the main operating system. It receives control from the system BIOS, performs a sequence of events, and then transfers control to the operating systems kernel.
- **Runlevel** A specialized script that starts a different set of services, permitting multiple configurations in the same system.
- **Kernel** The core operational code of an operating system. In Linux, it integrates the CPU architecture and supports the loading of modules and instructions to implement all operating system services (for example, process management, concurrency, and memory management).

INTRODUCTION

The Linux+ certified professional needs to have a good understanding of the overall Linux boot process. This knowledge is required to successfully install the Linux operating system, modify boot configurations, execute different runlevels, and troubleshoot boot issues. The Linux boot process, more complex than most operating systems, is based on four stages. The four stages are powering-up your system, loading and executing GRand Unified Bootloader (GRUB), loading and executing the Linux kernel, and loading the root. Figure 4.1 presents the four stages. For the Linux+ exam, this diagram can assist you in understanding how the Linux boot process works.

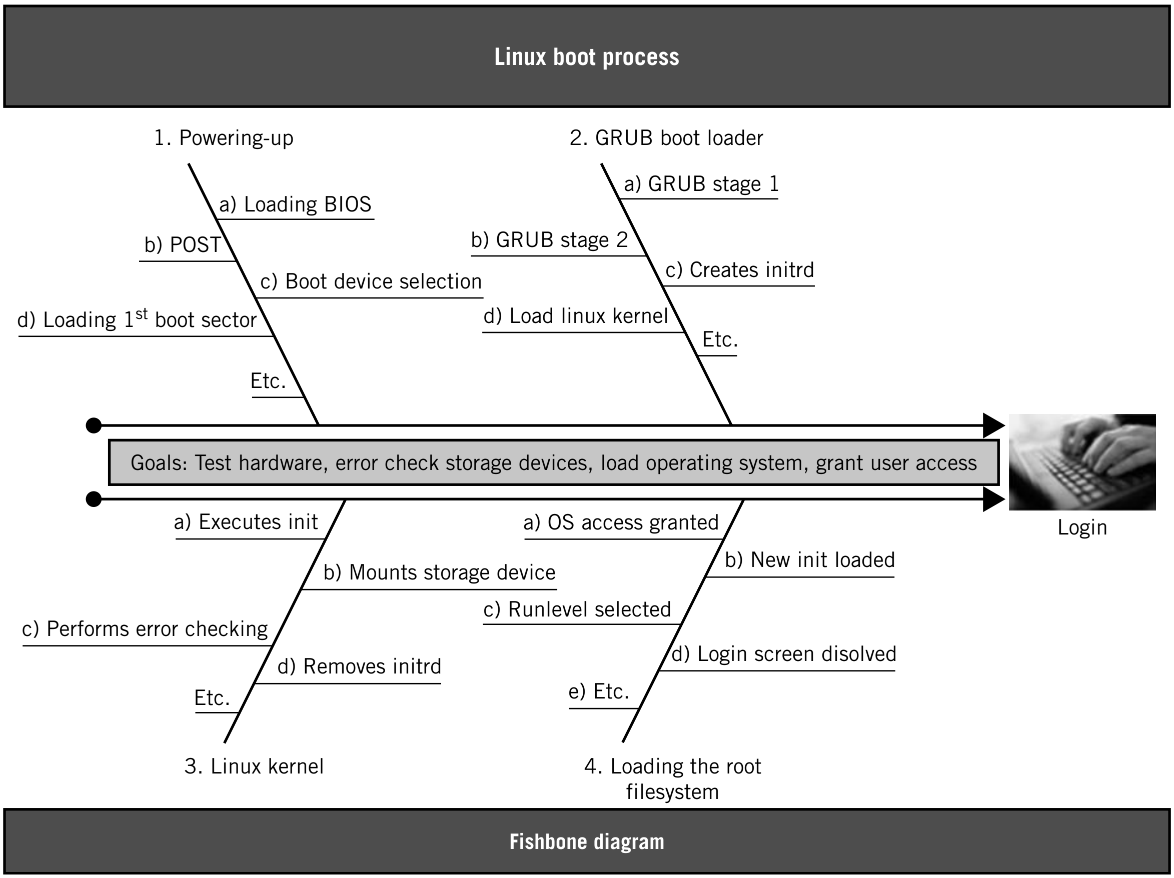


FIGURE 4.1 *Linux boot process model.*

The first stage, "powering-up your system," commences with the applying of electrical power to your targeted system. This normally entails pushing in the **power on/off** button. This process starts the initiation of the system BIOS. The system BIOS performs three different subtasks. The first subtask, *power-up self test* (known as *POST*), identifies, tests, and initializes critical system components such as the hard and floppy disk drives, random access memory (RAM), keyboard, video display card, and other hardware. The system BIOS, in addition, loads the system date and time.

Exam Warning

For the Linux+ exam, the Linux booting process is based on the Intel x86 CPU architecture. Linux also supports the booting of other CPU hardware architectures (for example, AMD, Alpha, ARM, IA-64, m68k, MIPS, PA-RISC, PowerPC, S/390, SPARC), but the booting processes are different.

The second subtask, *boot device selection*, determines what device will be used to boot the operating system. For this subtask, system BIOS is able to select from various devices for booting (for example, floppy disk drive, hard disk drive, CD/DVD drive) the Linux operating system. The system BIOS selects the first drive and loads the disk geometry characteristics (for example, cylinders, heads, and sectors).

In the third subtask, *boot sector loading*, the system BIOS reads the first sector of the boot device. This sector is 512 bytes in size. For a hard disk drive, this special sector is known as the *master boot record* (MBR). This sector contains the Linux bootloading program. The Linux operating system functions with various boot loaders (for example, LILO, GRUB, NTLOADER). openSUSE uses the GRUB program. The system BIOS loads GRUB into memory and executes the program. The system control is now transferred to the boot loader.

The second stage, "GRUB bootloader," performs two subtasks. For the first subtask, GRUB loads the Linux kernel (vmlinuz-version.gz). For the second subtask, GRUB creates and loads a virtual file system in system memory (RAM). This virtual file system is called the *initial ramdisk* (initrd). The initrd image contains various programs that are used to perform several tasks. This includes the "`init`" program and the necessary hardware drivers the Linux kernel will need to access storage drives. The init program contained in the initrd file is a bash script that loads the needed kernel modules.

The third stage uses the Linux kernel to execute the init program. The Linux kernel is an architecture designed to function dynamically. It supports

Table 4.1 Dual Roles of `init` Command

Existing Linux init Role	New Installation init Role
1. Loading the source containing the Linux distribution installation medium for installation	1. System drivers loading for Linux kernel
2. Hardware scanning and selecting of hardware drivers	2. Creating special system files for the kernel
3. Loading the installation system or rescue system	3. Managing redundant array of inexpensive disks (RAID) and Logical Volume Manager (LVM) setups
4. Linux installation package and system configuration tool	4. Managing network configuration

the loading of modules and instructions to implement all operating system services (for example, process management, concurrency, and memory management). The Linux kernel integrates the CPU architecture through a series of device drivers and kernel extensions. The init program will perform one of two different options, as shown in Table 4.1. The first option is the execution of a preinstalled (existing) copy of the Linux operating system. The second option is for a new installation of the Linux operating system.

More details about the `init` command will be provided later in the "Installing GRUB and Booting Linux" section. Regardless of which approach init takes, the init program mounts a mass storage device to obtain access to the root filesystem. Before granting access, the init program performs error checking. After the error checking process, the init program removes the virtual disk file system (initrd) from the memory.

The fourth stage, "loading the root filesystem," is the final stage. During this stage, the Linux kernel will use the init program to load and grant access to the actual root filesystem. This entails making sure the Linux kernel has access to the necessary hardware device drivers and the execution of any specific instructions for starting the Linux operating system (known as *runlevels*). More details about runlevels will be provided later in the "Runlevels" section. To perform this task, after making sure the Linux kernel has access, the original init program will execute a new init program that resides on a mounted hard drive. This new "`init`" program will load the Linux Login screen, as shown in Figure 4.2.

The four stages provide an overview of the Linux booting process for the Linux+ certified professional. The remaining focus of the chapter is on GRUB, runlevels, and troubleshooting boot issues.

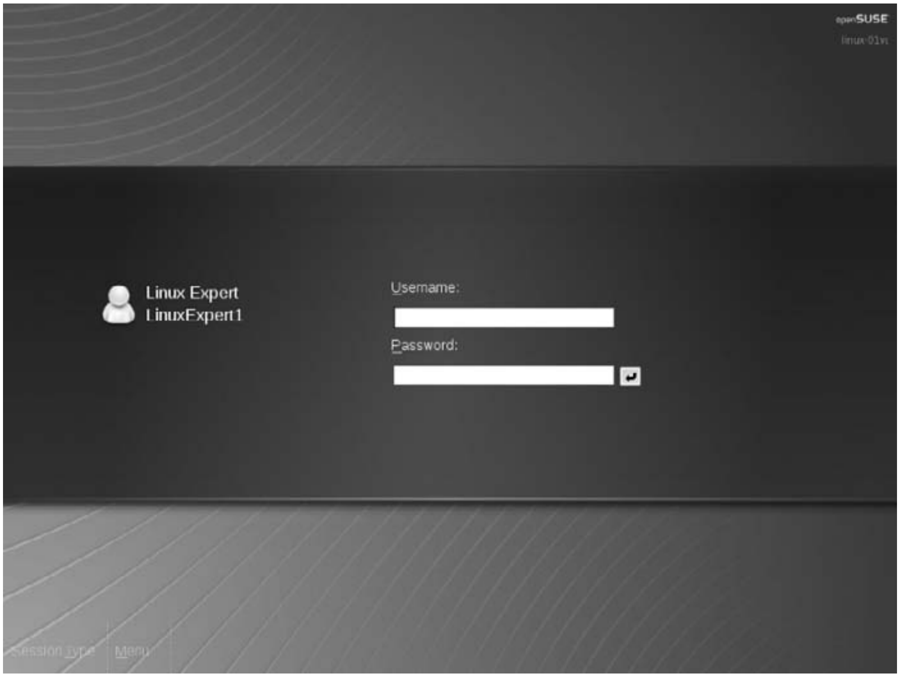


FIGURE 4.2
Linux login screen.

GRUB

Mentioned during the second stage, GRUB used for starting modern Linux operating systems, is the first program on any storage device that the computer executes. Before the bootloader, the system BIOS performed all operations on the system. The purpose of the bootloader program is to perform a sequence of events on your computer to load the main operating system. In essence, the bootloader receives control from the system BIOS process, performs a sequence of events, and then transfers control to the operating system kernel.

Most modern bootloaders are dynamically configurable and can be executed during the booting of the system and after the system has been booted. The bootloader can load predefined configuration files during startup or can support boot-time changes (for example, selecting different kernels, virtual file systems) through a boot command line prompt. In addition, the bootloader application can make modifications to boot-time configuration files and test the files before using them after control has been transferred to the operating system kernel.

GRUB supports both forms of use. This section presents GRUB in both forms. The first form is the use of GRUB to boot a Linux operating system. The second form is the use of GRUB to test and make modifications after system control has been transferred to the operating system kernel.

Installing GRUB and Booting Linux

Before the execution of GRUB, the system BIOS loads into memory the MBR and executes its contents. The total size of the MBR is 512 bytes. Most MBRs contain the bootloader program and disk partitioning information. Within the Linux environment, the preinstallation form of the GRUB program is divided into two stages. The MBR loads GRUB stage 1. The GRUB stage 1 program utilizes the first 446 bytes. The remaining 64 bytes are allocated to the partition table for the partitioning of the hard disk drives (for example, primary partitioning). The purpose of GRUB stage 1 is to find and load GRUB stage 2, which may reside physically elsewhere on the hard disk. To access the GRUB stage 2 program, GRUB stage 1 must be flexible enough to access many different file system types (for example, ext2, ext3, ReiserFS, and FAT). This also includes the ISO 9660 file system used for CDs or DVDs. This is accomplished because GRUB stage 1 has loaded a large number of mass storage device drivers. Once loaded, GRUB stage 2 can perform the following three different functions:

- It can load a predefined Linux kernel (for example, vmlinuz-version.gz).
- It can allow you to select which operating system to boot, if the computer was configured to boot multiple operating systems (for example, Windows, Linux).
- It can prompt you to enter different boot parameters.

Once GRUB stage 2 has loaded the Linux kernel (for example, vmlinuz-version.gz), it must also load a virtual file system and execute the Linux kernel.

Note

Some Linux distributions will use GRUB stage 1 to either load GRUB stage 2 directly or load a GRUB stage 1.5 application. If a GRUB stage 1.5 application is implemented into the boot process architecture, it is typically located in the first 30 KB of the hard disk immediately following the MBR. Once GRUB stage 1.5 is loaded, it will load GRUB stage 2.

GRUB Configuration Files and Commands

GRUB is a dynamically configurable bootloader application. This allows you to make postinstallation changes to your system for ensuring a successful boot-up process if device changes or Linux kernel modifications are required. To make changes to your system, the GRUB application allows you to make the alterations to three important configuration files.

The first file, /etc/grub.conf, contains information about the disk partition used to find and load GRUB stage2, as shown in Figure 4.3. This file instructs GRUB stage 1 where to look for the GRUB stage 2 image (/boot/grub/stage2) for loading.

In Figure 4.3, GRUB stage 1 will automatically install the GRUB stage 2 image located on the second partition of the first hard drive to the same partition and same drive. In addition, the configuration file instructs GRUB stage 1 to ignore any faulty logical block addressing issues by forcing the GRUB stage 2 installation process to continue. The GRUB configuration file location can vary across different Linux systems.

The second file, /boot/grub/menu.lst, functions as the GRUB Boot Menu. It contains content about the partitions and operating systems that can be booted, loading of different kernels, establishing of a different default kernel, and various other boot option modifications, as shown in Figure 4.4. The "Troubleshooting Boot Issues" section, presented later in this chapter, provides GRUB prompt procedures for entering commands that can be issued to dynamically modify the Linux kernel loading and runlevel processes.

The /boot/grub/menu.lst file options used for modifying and/or selecting a different kernel and various other bootloading functions are presented in Table 4.2.

The final configuration file, /boot/grub/device.map, is a unique file that maps the Linux device names to the GRUB/BIOS device naming conventions. Figure 4.5 presents the device.map file naming conventions for Integrated Drive Electronics (IDE) devices.

```
linux-01vc:/etc # cat grub.conf
setup --stage2=/boot/grub/stage2 --force-lba (hd0,1) (hd0,1)
quit
linux-01vc:/etc #
```

FIGURE 4.3
/etc/grub.conf file.

FIGURE 4.4
/boot/grub/menu.lst file.

```
linux-01vc:/boot/grub # cat menu.lst
# Modified by YaST2. Last modification on Mon May 11 12:50:48 UTC 2009
default 0
timeout 8
##YaST - generic_mbr
gfxmenu (hd0,1)/boot/message
##YaST - activate

###Don't change this comment - YaST2 identifier: Original name: linux###
title openSUSE 11.1 - 2.6.27.7-9
    root (hd0,1)
    kernel /boot/vmlinuz-2.6.27.7-9-pae root=/dev/sda2 resume=/dev/sda1 splash=silent showopts vga=0x317
    initrd /boot/initrd-2.6.27.7-9-pae

###Don't change this comment - YaST2 identifier: Original name: failsafe###
title Failsafe -- openSUSE 11.1 - 2.6.27.7-9
    root (hd0,1)
    kernel /boot/vmlinuz-2.6.27.7-9-pae root=/dev/sda2 showopts ide=nodma apm=off noresume nosmp maxcpus=
off powersaved=off nohz=off highres=off processor.max_cstate=1 x11failsafe vga=0x317
    initrd /boot/initrd-2.6.27.7-9-pae

###Don't change this comment - YaST2 identifier: Original name: floppy###
title Floppy
    rootnoverify (fd0)
    chainloader +1
linux-01vc:/boot/grub #
```

Table 4.2 /boot/grub/menu.lst Options

/boot/grub/menu.lst Options	Purpose
Default	This option is used to instruct the system to use the designated title entry to boot by default. Examples include the following: • default 0 – for the first menu • default 1 – for the second menu • default 2 – for the third menu
Timeout	This option is to instruct the system to immediately boot the default selection or wait a prescribed amount of time. Examples include the following: • timeout 5 – means wait 5 seconds before automatically booting the system • timeout 10 – means wait 10 seconds before automatically booting the system • timeout 0 – means boot the default selection immediately
Title	This option indicates the setting displayed by the boot-menu title. Example include the following: • title Linux • title Failsafe • title GNOME User Interface

(Continued)

Table 4.2 *(Continued)*

/boot/grub/menu.lst Options	Purpose
Root	This option provides a device or partition name indicating the location of the kernel and initrd files. Examples include the following: • root (hd0,0) – represents the first hard drive and the first partition • root (hd0,1) – represents the first hard drive and the second partition
Kernel	This option presents the location and name for the Linux kernel (for example, vmlinuz). This is the option to use to select a different kernel to boot the system. It also specifies the default runlevel by placing the runlevel number at the end of the line. Examples include the following: • kernel/vmlinuz-*version* • kernel/boot/vmlinuz-*version*
Initrd	This option provides the name and location of the virtual file system. This is the option to use to select a different virtual file system. Examples include the following: • initrd/boot/initrd • initrd/initrd
root = /*disk partition*	This option instructs the system where to mount the Linux root (/) directory. This is the option to use if a different device and/or partition is used for the root (/) directory. Examples include the following: • root = /dev/sda2 • root = /dev/sdb1
showopts	This option is used to display parameters listed after this option on the boot screen. Example: • showopts acpi=off ide=nodma

FIGURE 4.5
/boot/grub/device.map file.

GRUB, in the second form, is as an executable program (known as GRUB Shell) accessible from the root prompt. This executable program offers the following options:

- To change the disk order
- To view other boot loaders
- To view the hard disk partition details
- To modify partition settings
- To boot user-defined configuration files
- To password protect the system during the bootloading process

GRUB, in the second form, emulates the first form of GRUB, and you can install or test GRUB configuration settings before applying the modifications to the system during the next boot process.

EXERCISE 4.1: Establishing a Boot Password for GRUB

The Linux system grants anyone access to your system files during the GRUB boot process. To prevent an unauthorized person from accessing files on your Linux system during the boot process, a boot password needs to be assigned. In this exercise, we will assign a boot password to the system during interactive mode. To password protect other menu items; the "lock" keyword must be entered. You can also use the "password" command to provide a unique password for each menu item.

Complete the following:

1. Sign into the system using administrator (root) privileges.
2. From the root command prompt, type the `grub` command to enter an encrypted password: `# grub-md5-crypt`.
3. Press the **Enter** (Return) key.
4. The system will display *Password*: Enter ***<New password>***.
5. The system will display *Retype password*: enter ***<Retype New password>***.
6. The system will display *Encrypted: $1$59aS3/$/irAjfiPOy/hAwnB51ntg1* (only with your encrypted password).
7. Copy and paste the above encrypted string into the global section of the menu.lst file. The entry should look like the following:

```
password -m5: $1$59aS3/$/irAjfiPOy/hAwnB51ntg1
```

8. To execute any `grub` commands, you must press **P** and then enter the password. ■

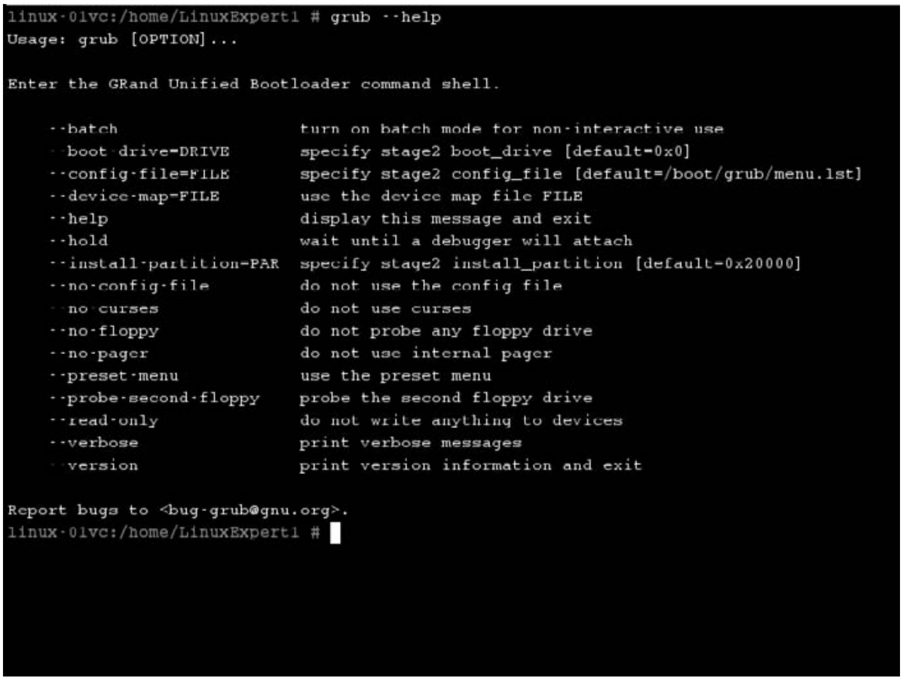
Figure 4.6, launched using root privileges, presents the `grub` help commands and the syntax format for the various options. The options presented can be executed within the GRUB utility or directly from the Linux prompt.

To implement changes made to the GRUB configuration device.map file, from the Linux prompt, execute the following command to reload device.map and execute the commands listed in the grub.conf file.

```
grub-batch < /etc/grub.conf
```

In rare occasions, you may be required to reinstall the GRUB preinstallation application to the hard disk on a running system. To accomplish this task, the `grub-install` command can be used, as shown in Figure 4.7. This command installs GRUB to either the MBR or another partition and checks for errors.

```
grub-install /dev/sda
```



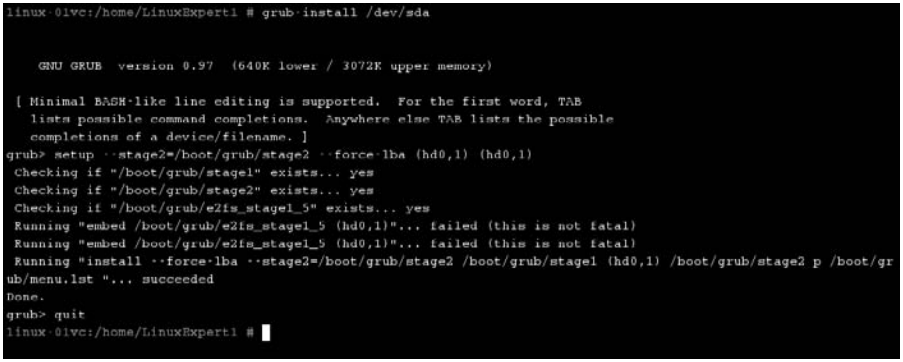
```
linux-01vc:/home/LinuxExpert1 # grub --help
Usage: grub [OPTION]...

Enter the GRand Unified Bootloader command shell.

    --batch                  turn on batch mode for non-interactive use
    --boot-drive=DRIVE       specify stage2 boot_drive [default=0x0]
    --config-file=FILE       specify stage2 config_file [default=/boot/grub/menu.lst]
    --device-map=FILE        use the device map file FILE
    --help                   display this message and exit
    --hold                   wait until a debugger will attach
    --install-partition=PAR  specify stage2 install_partition [default=0x20000]
    --no-config-file         do not use the config file
    --no-curses              do not use curses
    --no-floppy              do not probe any floppy drive
    --no-pager               do not use internal pager
    --preset-menu            use the preset menu
    --probe-second-floppy    probe the second floppy drive
    --read-only              do not write anything to devices
    --verbose                print verbose messages
    --version                print version information and exit

Report bugs to <bug-grub@gnu.org>.
linux-01vc:/home/LinuxExpert1 #
```

FIGURE 4.6
`grub` *help command.*



```
linux-01vc:/home/LinuxExpert1 # grub-install /dev/sda


    GNU GRUB  version 0.97  (640K lower / 3072K upper memory)

 [ Minimal BASH-like line editing is supported.  For the first word, TAB
   lists possible command completions.  Anywhere else TAB lists the possible
   completions of a device/filename. ]
grub> setup --stage2=/boot/grub/stage2 --force-lba (hd0,1) (hd0,1)
 Checking if "/boot/grub/stage1" exists... yes
 Checking if "/boot/grub/stage2" exists... yes
 Checking if "/boot/grub/e2fs_stage1_5" exists... yes
 Running "embed /boot/grub/e2fs_stage1_5 (hd0,1)"... failed (this is not fatal)
 Running "embed /boot/grub/e2fs_stage1_5 (hd0,1)"... failed (this is not fatal)
 Running "install --force-lba --stage2=/boot/grub/stage2 /boot/grub/stage1 (hd0,1) /boot/grub/stage2 p /boot/gr
ub/menu.lst "... succeeded
Done.
grub> quit
linux-01vc:/home/LinuxExpert1 #
```

FIGURE 4.7 `grub-install/dev/sda.`

RUNLEVELS

Today's modern Linux operating system can be used in various different ways for the user and administrator. This approach allows the Linux operating system to provide different services to be executed with limited and/or complete control of the system for the root administrator. This is needed to install, repair, and provide different levels of system maintenance without impacting the Linux user or preventing the execution of functions because users are still logged on and using the system. This section introduces the functionality of the init program and the seven different Linux runlevels.

The `init` Command

The `init` command, as indicated in this chapter, provides two services. The first service is the version of `init` accessible by the Linux kernel after the virtual file system is created. It was discussed earlier in the "Introduction" section of this chapter for executing a new installation of a Linux distribution or executing an existing installation of a Linux distribution.

Regardless of which init program is used, both versions of the program are used by your Linux system. The second init, the one retrieved from the root (/) filesystem is responsible for executing start and stop Linux runlevels.

Linux Seven Runlevels

Runlevels are specialized scripts that define how a computer system starts or stops by executing various services or processes and the level of root administration and user access. The runlevel is changed by having a privileged user run `telinit` or `init`, which sends appropriate signals to init. For your

Linux system, seven different runlevels exist. The runlevels are numbered from 0 through 6. Table 4.3 provides a description of each of the runlevels.

Your system can be configured to launch any of the above runlevels. Some choices, runlevels 0, 4, and 6 are obvious choices you do not want to configure as initial startup runlevels. However, the other runlevels are viable options depending on your objectives. Most systems set their default runlevels to either 3 or 5.

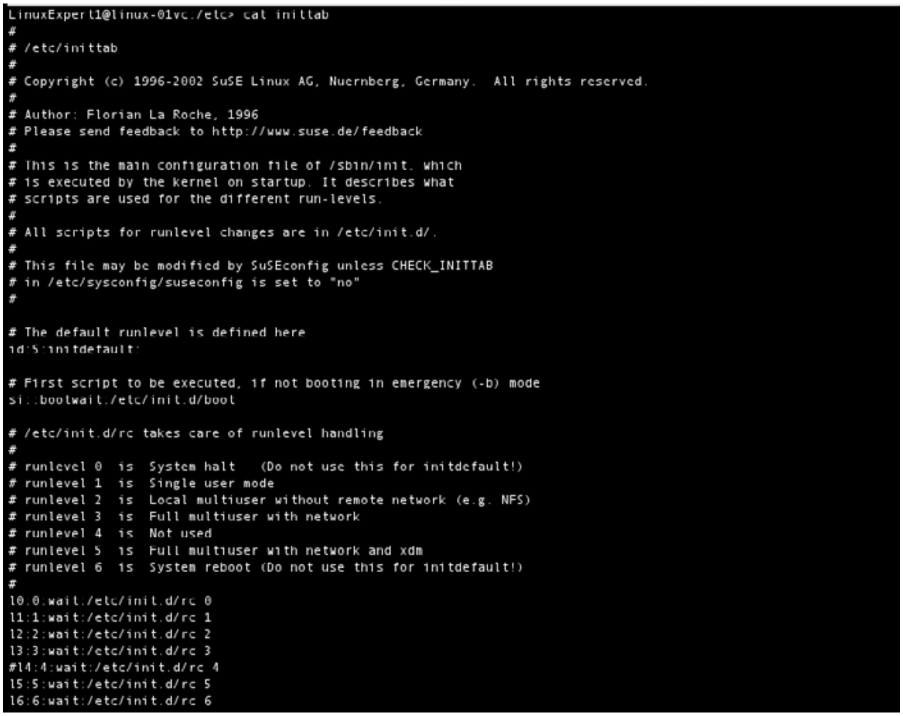
To execute runlevels, two types of scripts are located in the /etc/init.d directory and both types are called through *symbolic links*. The first set of scripts, executed by the `init` command, is started during the system's boot process or whenever you initiate the shutdown process. These scripts are contained in the /etc/inittab file, as shown in Figure 4.8. The default runlevel setting is located in the file /etc/inittab. The default runlevel entry looks like the following in the file:

```
id:5:initdefault:
```

For the file presented in Figure 4.8, the runlevel is configured to start up in multiuser mode and provide network connectivity and X Windows support for Windows Managers. In addition to the default runlevel listed in the file,

Table 4.3 Runlevels

Runlevel	Description	Special Comments
0	This runlevel shuts down or halts your system.	This option terminates all programs and services. Be careful.
1	Single user mode	The parameters s or S can be used as substitutes for 1.
2	Functions in multiuser mode, but there is no network connectivity. Login locally is the only option available.	If your system mounts a network file system (for example, SAMBA, NFS), never use this option.
3	Functions in multiuser mode with full network connectivity	This starts the system up in command line (terminal) mode. There is no graphic interface.
4	This option is user configurable, but normally is undefined	Not used
5	Functions in multiuser mode, but it also provides network connectivity and X Windows support for Windows Managers	This starts the system up in graphical user interface mode (for example, GNOME, KDE).
6	This runlevel reboots your system.	This option terminates all programs and services. Be careful.



```
LinuxExpert1@linux-01vc:/etc> cat inittab
#
# /etc/inittab
#
# Copyright (c) 1996-2002 SuSE Linux AG, Nuernberg, Germany.  All rights reserved.
#
# Author: Florian La Roche, 1996
# Please send feedback to http://www.suse.de/feedback
#
# This is the main configuration file of /sbin/init, which
# is executed by the kernel on startup. It describes what
# scripts are used for the different run-levels.
#
# All scripts for runlevel changes are in /etc/init.d/.
#
# This file may be modified by SuSEconfig unless CHECK_INITTAB
# in /etc/sysconfig/suseconfig is set to "no"
#

# The default runlevel is defined here
id:5:initdefault:

# First script to be executed, if not booting in emergency (-b) mode
si::bootwait:/etc/init.d/boot

# /etc/init.d/rc takes care of runlevel handling
#
# runlevel 0  is  System halt   (Do not use this for initdefault!)
# runlevel 1  is  Single user mode
# runlevel 2  is  Local multiuser without remote network (e.g. NFS)
# runlevel 3  is  Full multiuser with network
# runlevel 4  is  Not used
# runlevel 5  is  Full multiuser with network and xdm
# runlevel 6  is  System reboot (Do not use this for initdefault!)
#
l0:0:wait:/etc/init.d/rc 0
l1:1:wait:/etc/init.d/rc 1
l2:2:wait:/etc/init.d/rc 2
l3:3:wait:/etc/init.d/rc 3
#l4:4:wait:/etc/init.d/rc 4
l5:5:wait:/etc/init.d/rc 5
l6:6:wait:/etc/init.d/rc 6
```

FIGURE 4.8
Linux inittab scripts file.

the file also indicates what subdirectory to search for specific services or programs to execute. For the default runlevel, /etc/init.d/rc5.d is the directory containing default services and programs scripts. Each of the runlevel folders contain scripts that are executed when you start a runlevel and when you stop a runlevel. For files containing scripts used for starting a runlevel, these files begin with the capital letter "S." For files containing scripts used for stopping a runlevel, the files begin with the capital letter "K." The files containing S are used for starting a process and the files containing K is for killing a process. The second sets of scripts are executed whenever the runlevels are changed from one runlevel to another runlevel. The /etc/init.d/rc file is called to ensure the scripts are executed in the proper sequence.

> **Note**
> The Linux kernel starts the init process. This process is assigned Process ID 1. All subsequent processes are parent processes or child processes launched from the init process.

EXERCISE 4.2: Changing to the Single User Mode (Runlevel 1)

For this exercise, you will change from your current runlevel to single user mode (runlevel 1). This runlevel allows you to perform system administration tasks without network access/graphic interface functionality. *(Always make sure all users are logged off before performing this function.)*

Perform the following:

1. Login as system administrator (root) by using the switch user command `su`.
2. From the root command prompt, enter `init 1` at the root prompt.
3. Press the **Enter** (Return) key.
4. The system will commence the process of executing stop scripts for your current runlevel and then commence executing start scripts for single user mode (runlevel 1).
5. Login as the root user.
6. Enter the root user password.
7. Now you can perform various system administrative tasks without impacting local or network users. ■

TROUBLESHOOTING BOOT ISSUES

In life, it would be nice if all things performed as they should. However, within the IT profession Murphy's Law still exists. For the Linux+ exam, the Linux professional must be able to troubleshoot and resolve problems that may occur during the Linux boot process. This section provides an overview of different options available to the Linux professional to resolve Linux booting issues.

The first option is the `dmesg` command, which is used to send Linux kernel messages to a standard output (for example, computer monitor). After the completion of GRUB, the bootloader, the Linux kernel is loaded and executed. During this phase, the kernel can send messages to the computer monitor representing hardware devices detected and if it is able to configure the devices. `dmesg` accomplishes this by being able to print or control the kernel ring buffer. For the Linux professional, `dmesg` can assist in troubleshooting or obtaining information about system hardware. The `dmesg` syntax is as follows:

```
dmesg [ -c ] [ -n level ] [ -s bufsize ]
```

- The `-c` option clears the kernel ring buffer.
- The `-s` option, which includes buffer size, determines the buffer size to query from the kernel ring buffer. The default buffer size is 16,392 by default.
- The `-n` option, which includes the logging level, determines the type of messages sent to the computer monitor. When "`-n 1`" is used, only panic messages appear on the computer monitor. All other messages are prevented. The `dmesg` command will not print or clear the kernel ring buffer when the `-n` option is used.

When entering the `dmesg` without any parameters, the computer monitor will not be able to display the entire list of kernel messages sent to the standard output. If the messages scroll passed the screen too quickly, you can enter the `dmesg | more` command to review the messages one page at a time.

The second option, *kernel options*, allows you to enter information to be executed by the Linux kernel during the booting of your system. The information passed to the kernel allows you to control the behavior of your system when it boots up. The kernel parameters can be added by editing /boot/grub/menu.lst or by entering information at the boot prompt. The /boot/grub/menu.lst provides the default kernel parameters, as shown earlier in Figure 4.4. For the /boot/grub/menu.lst file, kernel parameters can be added or modified. In addition, fail-safe kernel parameters can be predefined that will enable Linux to boot even under problematic circumstances.

To dynamically modify the kernel during the boot process with GRUB, you can enter various kernel parameters. The most common kernel parameter is the modification of the runlevel parameter. Figure 4.9 presents the entering of runlevel 1 into the *Boot Options* field, during system startup to boot the system into the single user mode.

The third option, *rescue system*, starts a specialized Linux kernel without a graphical user interface. It can be loaded from various sources and locations. The easiest approach is to use the original Linux distribution to boot the system. The Linux kernel loaded can be obtained from a CD, DVD, or any bootable device. Once the Linux kernel is loaded into RAM, you can modify configuration files, check the file system for defects, verify and/or modify the bootloader configuration, resize the partition, and a few other critical system modifications that may be necessary. Many of the Linux distributions are also available in Live CD form (Knoppix, openSUSE LiveCD, Fedora LiveCD) as well. This form allows you to perform a system rescue by booting the system and then mounting the disk partitions to fix or repair configuration or boot issues.

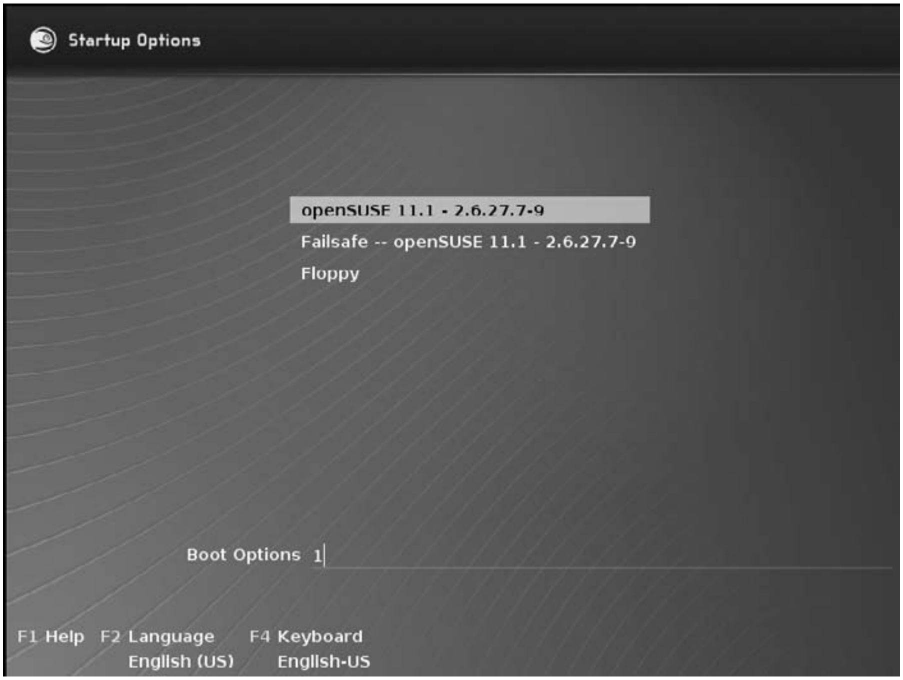


FIGURE 4.9
Boot Options *for the Linux kernel.*

Learn By Example: To Be or Not To Be . . . That is the Question!

A friend of mine, who was recently promoted to CIO of a midsize company, was very concerned about the security of its Linux servers. The system administrator would brag, in front of my friend, about how secure its Linux servers were and that no one could gain access to the proprietary company files on the hard drives. The system administrator installed all the normal technical security controls (for example, firewalls, IDS, strong encrypted passwords, antivirus software) one would expect. The system administrator felt a security assessment was not necessary.

My friend, with valid concerns, hired me to perform a series of security penetration tests. He wanted me to assess both the company's local and remote security defenses. I asked the customer to give me about 10 minutes alone with the system to perform a common physical security attack. The system administrator logged off of the Linux system after checking to make sure I could not penetrate the company's security defenses. They left to get a few snacks from the vending machine. Within 5 minutes, I rendered the system defenseless. I change the root password and gave the customer copies of critical company files.

How did I obtain access so quickly, you ask? I forced a reboot of their Linux servers and obtained access because there was no bootloader (GRUB) password. I was able to load any kernel I wanted, change boot partitions, and access critical files. Remember,

technical security controls are never enough to prevent an insider from physically penetrating your security defenses. Physical and administrative security controls are also important.

The fourth option, *single user mode*, is used to bypass the requirement to enter a root password (many Linux distributions now force the entering of a root password). Typically, this option is used to gain access to the root prompt to change a lost or forgotten password. For this option, you will boot the Linux system under runlevel 1 (single user mode), and you will directly get a root prompt. For this level, you can execute the `passwd` command to modify the root password. Since you are logged in as root, you will not be prompted to enter the old password.

Note
Many of the Linux distributions today hide the boot messages, normally displayed on the screen by a splash screen. However, you can view the boot messages by pressing the **ESC** key when the splash screen appears.

EXERCISE 4.3: Lost or Forgotten Root Password on Linux Server

In this exercise, we will use the rescue system to obtain root-level system access to the system and change the password.

Complete the following:

1. From a Linux bootable installation source media, boot the target machine.
2. This will start the openSUSE Installer. Select **Rescue System**.
3. The system will display *Rescue login*; Enter **root**.
4. No password is required. This can produce a security concern, if no physical security is implemented for the target system. The system will display the `Rescue:~#` command.
5. At the prompt enter: `fdisk -l` (to list the partitions).
6. Locate the root partition.
7. Type `mount /dev/sda1 /mnt`. (Replace sda1 with the device name for the Linux root partition identified in the previous step.)
8. Type `mount -o bind /dev /mnt/dev?` (to make the device files available).

9. Type `chroot /mnt`.
10. Type `passwd root` (to reset the root password).
11. Enter new root password: < >
12. Type `exit` (to leave the chroot environment).
13. Type `umount /mnt` (to unmount the filesystem).
14. Reboot the system.

SUMMARY OF EXAM OBJECTIVES

In this chapter, we discussed the Linux boot process and the information you will be required to know during the booting of a Linux system. The Linux boot process was presented in four stages. The four stages were the powering-up of your system, loading and executing GRUB, loading and executing the Linux kernel, and loading the root filesystem. After presenting an overview of the Linux boot process, details were presented for three key components of the Linux boot process.

For the first section, GRUB, the default bootloading application was introduced. GRUB's purpose is to perform a sequence of events on your computer to load the main operating system. In essence, the bootloader receives control from the system BIOS process, performs a sequence of events, and then transfers control to the operating system kernel. The GRUB program was presented in two forms. The preinstallation form normally used during the initial booting of your system, and the postinstallation form normally used as an application to install or test GRUB configuration settings before applying the modifications to the system during the boot process.

For the preinstallation form, GRUB stages 1 and 2 were introduced to describe how the Linux kernel and mass storage devices are accessed. For the postinstallation form, three critical GRUB configuration files were presented. The /etc/grub.conf, /boot/grub/menu.lst, and /boot/grub/device.map are the three files. Each of the three configuration files are used during the Linux boot process, however, it is during postinstallation that the files are typically modified and tested by the GRUB shell command.

In addition, during the postinstallation two GRUB commands are presented. The two commands are `grub-batch` and `grub-install`. The `grub-batch` command is used to implement changes made to the GRUB configuration device.map file and reloads the device.map file before executing commands listed in the grub.conf file. The `grub-install` command is used to reinstall the GRUB preinstallation application to the hard disk

on a running system. This command installs GRUB to either the MBR or a partition and checks for errors.

For the second section, the `init` command and the Linux runlevels were introduced. The `init` command executes the runlevels. The `init` command, as indicated in this chapter, provides two services. Like GRUB, `init` functions in a preinstallation and postinstallation forms.

In the postinstallation form, the `init` command is located on the root (/) filesystem. This version of init also provides four functions. The first function is the loading of the correct file system drivers for the kernel to load the root (/) filesystem. The second function creates special system files for the filesystem and other system drives. The third function is used to configure and grant access to redundant array of inexpensive disks (RAID) and LVM functionality. The fourth function provides assistance for loading network drivers to support network mounted filesystems (for example, SAMBA, NFS).

Runlevels use a collection of scripts that define how a computer system starts or stops by executing various services or processes and the level of root administration and user access. The runlevels are numbered from 0 through 6. Each level provides or removes a level of functionality for the user and the system. For example, no network access, no graphic user interface, or single user mode only. The default runlevel settings for the system are located in the /etc/inittab file.

The final section, "Troubleshooting Boot Issues," introduces the Linux+ professional to four different approaches to resolve the boot problems. The `dmesg` command, dynamically modifying the Linux kernel during the boot process, using system rescue procedures, and entering single user mode to regain root access are the four approaches.

For the Linux professional, `dmesg` can assist in troubleshooting or obtaining information about system hardware. The `dmesg` command is used to send Linux kernel messages to a standard output (for example, computer monitor). `dmesg` accomplishes this by being able to print or control the kernel ring buffer.

The dynamic modification of the kernel option allows you to enter information to be executed by the Linux kernel during the booting of your system. The kernel parameters can be added by editing /boot/grub/menu.lst or by entering information at the boot prompt or by dynamically modifying the kernel during the boot process with GRUB. The `runlevel` parameter is the most common kernel modification.

The system rescue approach starts a specialized Linux kernel without a graphical user interface. The Linux kernel loaded can be obtained from a CD, DVD, or any bootable device (for example, LiveCD). Once the Linux kernel is loaded into RAM, you can modify configuration files, check the

file system for defects, verify and/or modify the boot loader configuration, resize the partition, and a few other critical system modifications that may be necessary.

The single user mode is the final troubleshooting approach. Because of the security implications of this approach, this option is available in some Linux distributions. Unlike the previous approaches, this approach focuses on regaining access to the root account to change a lost or forgotten password. This approach entailed gaining access to the root prompt under runlevel 1 (single user mode) and changing the root password.

SELF TEST

1. You need to access your department's Linux server to perform system maintenance. To perform the necessary administrative tasks, all users need to be logged out of the system and they are not allowed to log back into the system while the system maintenance activities are underway. Which runlevel only grants root access?

   A. 6

   B. 0

   C. 2

   D. 1

2. Your department's manager would like all Linux users to access their workstations by using a graphical user interface and have network connectivity. Which runlevel uses a graphical user interface by default and grants network connectivity?

   A. 2

   B. 0

   C. 5

   D. 1

3. Your department's manager would like all Linux users to access their workstations by using a command line mode (no graphical user interface) and have network connectivity. Which runlevel uses a command line mode for multiple users and grants network connectivity?

   A. 2

   B. 0

   C. 3

   D. 1

4. What is the purpose of the computer system BIOS?
   - A. Loads the Linux kernel before loading Linux GRUB
   - B. Allows the user to log into the Linux operating system and change the kernel
   - C. Presents the biography of Linus Torvalds, the creator of Linux
   - D. Commences the Linux boot process
5. You need to access your department's Linux server to perform system maintenance. You need to power down the system to install new hardware components. Which runlevel shuts down your system?
   - A. 6
   - B. 0
   - C. 2
   - D. 1
6. The Linux servers in your department all use IDE hard disk drives. Your supervisor requested that you reinstall GRUB into the first partition on the IDE first hard disk while the machine is still running. To install GRUB on the IDE hard disk drive's first partition, which shell command should you use?
   - A. `grub ide`
   - B. `grub-install /dev/hda1`
   - C. `grub-install /dev/sda1`
   - D. `grub-runlevel /dev/hda1`
7. You are the Linux system administrator for your IT department. When you normally access your workstation in the morning, you are granted multiuser access with graphical user interface and network connectivity. To perform system maintenance activities, you need to switch runlevels when the system is running. Which command is used to switch runlevels when the system is running?
   - A. `runlevels`
   - B. `init`
   - C. `System Rescue`
   - D. `grub`
8. How large is the Master Boot Record (MBR) for a hard disk drive with a sector size of 512 bytes?
   - A. 1 MB
   - B. 512 bytes

C. 0 bytes

D. 6 KB

9. Which order of events represents the proper Linux boot process?

A. System BIOS, bootloader, Linux kernel, user logs into the system

B. Bootloader, system BIOS, Linux kernel, user logs into the system

C. System BIOS, Linux kernel, bootloader, user logs into the system

D. User logs into the system, system BIOS, bootloader, Linux kernel

10. Which command is used to send Linux Kernel messages to the standard output (for example, computer monitor)?

A. `grub`

B. `dmesg`

C. `init`

D. `kernelprint`

11. You are the IT system administrator for the Linux systems in your department. You need to make changes to the default runlevel setting. Which file contains the default runlevel setting?

A. /etc/inittab

B. /etc/grub.boot/inittab

C. /boot/grub/device.map

D. /etc/init.d

12. Your IT department has made several hardware device changes. These changes include modifications to the hard disk drives. You need to make modifications to the GRUB bootloader. Which file should you edit to configure the GRUB stage 2 image?

A. /etc/menu.lst

B. /boot/grub/menu.lst

C. /etc/grub.conf

D. /boot/grub/grub.conf

13. You are the IT system administrator for the Linux systems in your department. You need to make changes to the GRUB device naming conventions. Which file contains the default runlevel setting?

A. /etc/device.map

B. /etc/grub.boot/device.map

C. /boot/grub/device.map

D. /etc/init.d

14. The Linux kernel is a critical component in the Linux boot process. Where does it reside on the system?
    - A. The /kernel directory
    - B. The /grub/boot/kernel directory
    - C. The /boot directory
    - D. The /boot/kernel directory
15. The Linux bootloader is a very critical component in the Linux boot process. Where does it reside on the system?
    - A. It resides in the Master Boot Record.
    - B. It resides inside the Linux kernel.
    - C. The /etc directory
    - D. Inside the system BIOS

SELF TEST QUICK ANSWER KEY

1. D
2. C
3. C
4. D
5. B
6. B
7. B
8. B
9. A
10. B
11. A
12. C
13. C
14. C
15. A

CHAPTER 5

Configuring the Base System

Exam objectives in this chapter

- User Profiles
- Device Management
- Networking

UNIQUE TERMS AND DEFINITIONS

- **Route** It is the path from a source device through a series of hosts, routers, bridges, gateways, and other devices that network traffic takes to arrive at a destination device on a different network.
- **Port (TCP/IP)** It is a logical channel or channel endpoint in a communications system. Each application program has a unique port number associated with it. Port numbers distinguish between different logical channels on the same network interface card (NIC).

INTRODUCTION

This chapter will explain how to configure system and user profiles, as well as the common environment variables; management of various devices and where these are located in the disk structure; and the fundamentals of

Linux networking utilizing Transmission Control Protocol/Internet Protocol (TCP/IP) and how to manage this within Linux.

User management is one of the fundamental tasks that needs to be understood for day-to-day management of a Linux system, and to be able to achieve this in an effective manner is necessary for all Linux administrators. This is equally important whether the system is used by yourself, your whole family, or a large corporation. Users should have the correct rights and environment setup to ensure that their experience in using the system is favorable and that the support overhead is kept to a minimum.

As part of the system setup, the user will need and want to access the various devices installed on the system. The correct setup of these devices needs to be accomplished prior to deployment to ensure that the system is functioning correctly. This will typically need to be undertaken at the superuser level, a level of privilege that is typically not bestowed onto a normal user. In addition, the user interaction at this level can often cause system instabilities, which may be hard to diagnose.

The networking of the computer system, whether via wired or wireless connection, is usually a given necessity in today's world. There will be very few systems deployed without this capability, and often, the user will require both options, particularly in any form of portable device. The computer will also typically use TCP/IP as the transport mechanism for the network connections, and the different options available to set up this will be discussed. The basics in connecting to name servers and Dynamic Host Configuration Protocol (DHCP) servers are also discussed, and the majority of parameters are explained.

USER PROFILES

Any installation of Linux will include the creation of a number of different user accounts: the superuser, a normal day-to-day user, and a system user. Each of these accounts is important in their own right, and a user needs to understand the differences.

All users need to have an account on the system, which should be unique to them to ensure that there is a basic level of security built in. A normal user can add, delete, and modify their files and those that have the appropriate attributes set. These users cannot make system wide changes nor can they manage other users on the system. Standard users can also make changes that are specific to them, such as desktop wallpaper and addition of printers.

Users who are specified as being a superuser (also referred to as a system administrator) have global privileges, can create and delete users, and can

change the permissions of files located within the filesystems. There may be many system administrators on a given system, particularly those located in a large corporation. The number of users who are elevated to this level should be kept to a minimum to ensure the security and integrity of the underlying system. There is a special superuser known as *root*, with a user ID and group ID of 0. This user has full and unrestricted rights to manipulate any file, to traverse to any directory, and to execute any program. For obvious reasons of security, this user's credentials should be shared by very few people.

The third type of account is that of a system user, which is really not a user at all. This account is an administrative account that is used by the system itself for the running of various administrative tasks. For example, xfs owns the X11 font server and all its associated files, and these can be executed only by itself and the root account. System users differ from other users on the system in that they do not have a home directory or password nor can they be accessed via the normal system login prompt.

System and User Profile and Environment Variables

The following will describe the user profiles that are created for each individual user and the common variables that are modified to suit your personal preference. Each user on the system is able to customize their profile to suit their specific needs and preferences. These preferences are held in environment variables located in resource files throughout the system.

The three different shells in bash are the login shell, the normal shell, and the interactive shell. The login shell reads the .profile file located in the user home directory or /etc directory (~/.profile), and interactive shells read ~/.bashrc. The environment variables are named objects that contain information that can be used by one or more applications.

As bash is the default shell under Linux, a summary of the bash startup files is shown below:

- /etc/profile This is the system-wide startup file and will be executed when a user logs on. The file will be protected, and only the superuser (root) will be able to make changes to the file. As this file may be overwritten with a system upgrade, it is not recommended that changes are made to this directly.
- /etc/bash.bashrc This is often linked to /etc/bashrc, and is called *per interactive shell startup*. The file will be protected, and only the superuser (root) will be able to make changes to the file. As this file may be overwritten with a system upgrade, it is not recommended that changes are made to this directly.

Normal users will have two similar files, which are also called as

- **/home/user/.bash_profile** This is a personal startup file and is executed when a user logs into a system.
- **/home/usr/.bashrc** This is the personal interactive shell startup file.

Both the above files can be edited by the user using an editor such as vi. The format for creating and modifying an environment variable within bash is always in the format given below:

```
NAME=value
```

This will define the variable in the shell only. To move that from the shell to the environment, the `export` command has to be used:

```
export NAME=value
```

This allows programs other than the shell to access this variable (for instance, a file editor).

PS1

When a user first logs on, they will be greeted by a prompt. This prompt can be changed and may be different depending on the version of Linux that has been installed. The typical prompt for a standard user is the $ symbol, while root is denoted by the # symbol. This can be demonstrated below (user input is in boldface).

```
$ who am i
syngress
$ su root
Enter password for root: xxxxxx
# who am i
root
# exit
$ who am i
Syngress
```

The user prompt defined initially in the file /etc/bash.bashrc as the environment variable `PS1` can display a vast array of data. The `PS1` variable can change the command line as shown below:

```
$ PS1="\u>"
syngress>
```

Typical variables that are used are given below:

`\d` is the date in "Wed Sep 09" format

\h is the first part of the hostname (such as mysystem)

\u is the username

\t is the time in 24-h format

The complete list of strings can be found in the documentation, including how to add colors and sounds (which may be useful to highlight the fact that you are operating at the root level; or, for system administrators, which machine you are actually logged onto). Modifying the variable as above will not make it permanent, and will revert back to the system setting when the user logs off and then back in again. To make the changes permanent, they have to be included in your local profile.

PS2

The PS2 variable is very similar to the PS1 variable, except that it is displayed when the user issues an incomplete command and the system will prompt and wait for the user to complete the command and press **enter** again. This default secondary prompt is the > sign and can be changed by altering the PS2 variable.

```
$ echo ‘‘this is a
> test’’
this is a
test
$
```

Redefining the PS2 variable with a customized prompt is shown below:

```
$ PS2=’’more input > ‘‘
$ echo ‘‘this is a
more input > test’’
this is a
test
$
```

> **Note**
> There are other prompt string variables available in some shells, such as PS3 and PS4 in the Korn shell. These are not specifically required in the exam due to their specific nature.

Path

The PATH variable is used by commands to locate a specific command or application. When you enter a command, the shell will look in each of the

directories specified in the PATH command, and will return a "command not found" message if the command is not located in one of those directories. Typically, if a user creates a command, it should be placed in the bin directory of their home directory (for example, /home/syngress/bin) or another convenient location, and the PATH variable checks to ensure that the directory is referenced. The PATH variable contains the list of directories separated by a colon, as shown below:

```
echo $PATH
/bin:/usr/bin:/usr/local/bin
```

As this does not contain your local bin directory, you will need to add it to the variable. The PATH variable can be modified to include the directory by issuing the following command:

```
PATH=$PATH:$HOME/bin
```

If PATH was defined as beforehand, executing the commands as before (for the user syngress) would result in the following:

```
echo PATH
/bin:/usr/bin:/usr/local/bin:/home/syngress/bin
```

Editor

The EDITOR variable defines the users preferred test editor. The number and range of text editors are vast, and each user has their own favorite (and dislikes!) A number of common ones are ed, vi, vim, and emacs. As mentioned above, the setting of the environment variable is achieved by setting on the command line or inserting into the .bashrc script.

The command echo $EDITOR will return a blank line if the variable is not set. To set the default editor to be vim, use EDITOR=vim or EDITOR=/usr/bin/vim.

Ideally, it is better to use the complete pathname (especially in the .bashrc script) to ensure that the variable is defined correctly. If the PATH variable is changed or does not have the /usr/bin directory defined, the editor will not be found and an error will ensue. How to use the vi text editor is defined in Chapter 6.

Term

The TERM variable is to set up the type of terminal in use, which can be particularly important when using screen-oriented programs such as a text editor. This variable is set automatically during system installation but may need to be modified if a user is remotely accessing the system. When Linux is installed with KDE, the TERM variable will be set to the value xterm.

The variable can also be set using the `tset` command, which is often used in the login script for a user to allow them to choose the type of terminal they are logging in from.

Pager

The `PAGER` variable controls the output to the screen, such as the `man` command. This allows the display of the output in a controlled manner. The typical values for `PAGER` are `more` and `less`. While these are similar, `less` has additional features (such as scroll backwards with the **b** key) and hence is the preferable value to use as a default.

Home

The `HOME` variable is set whenever you login to the system and will be set to /home/username (where username is your login name). This should not be changed, as a lot of programs use this to create or find files in your personal home directory. In addition, the shortcut ~ references the `HOME` variable, and will return an erroneous result if this is modified.

Printer

The `PRINTER` variable defines the default printer. If no printer is defined, this value will be set to `NULL`. This variable is mainly used for command-line programs to print. The setting of the variable does not preclude the need to install the printer on the system. The use of the `PRINTER` variable allows the system administrator to set up specific printers for users based on their location.

EXERCISE 5.1: Configuring the User Environment

In this exercise, you will be changing the default `PAGER` variable setup in your profile.

1. Change your directory to your home directory.
2. Perform a `more .bashrc` command to see your current environment setup. You see that the `PAGER` variable is set to `more`.
3. Type the command **`echo $PAGER`**, which shows the value `more`.
4. Using your favorite editor, add the following line to the *.bashrc file*

   ```
   export PAGER=less
   ```

5. This will change the value from `more` to `less` when the user next logs out and back in again. ■

DEVICE MANAGEMENT

The management of devices on a Linux system is critical, and at first glance can seem more complicated for users of Microsoft Windows getting to grips with Linux. There is often more than one way of managing a device, and often the command line can be used as well as one of many graphical user interfaces (GUIs). The following section will list some of the common commands and major directories that will likely be used on a day-to-day basis by a system administrator.

lsusb

`lsusb` will list all the usb buses on a system and display information about any devices attached to them. The `-v` and `-vv` options will give a verbose output; however, this will only be required during debugging as the output really is verbose, with the `-vv` option giving out all the information the peripheral component interconnect (PCI) device can display. This command is useful to see what usb devices are working and the type of device that is attached.

```
$ lsusb
Bus 004 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 003 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 002 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 001 Device 005: ID 0204:6025 Chipsbank Microelectronics Co., Ltd
CBM2080 Flash drive controller
Bus 001 Device 003: ID 046d:c517 Logitech, Inc. LX710 Cordless Desktop
Laser
Bus 001 Device 002: ID 413c:0058 Dell Computer Corp. Port Replicator
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

lspci

The `lspci` command is similar to the `lsusb` command, except that it displays all the information regarding the PCI buses in the system and all the devices that are attached to it. This command is useful to understand what devices are attached to the system and the actual driver that should be loaded for the devices. This command can also be used with the `-x` option to display the initial 64 bytes of PCI configuration, which can be useful to see what is loaded.

```
$ lspci -x
00:00.0 Host bridge: Intel Corporation 82855PM Processor to I/O
Controller (rev 03)
```

```
00: 86 80 40 33 06 01 90 20 03 00 00 06 00 00 00 00
10: 08 00 00 e0 00 00 00 00 00 00 00 00 00 00 00 00
20: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
30: 00 00 00 00 e4 00 00 00 00 00 00 00 00 00 00 00

00:01.0 PCI bridge: Intel Corporation 82855PM Processor to AGP
Controller (rev 03)
00: 86 80 41 33 07 01 a0 00 03 00 04 06 00 20 01 00
10: 00 00 00 00 00 00 00 00 00 01 01 20 c0 c0 a0 22
20: 00 fc f0 fd 00 e8 f0 ef 00 00 00 00 00 00 00 00
30: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0c 00
```

In addition, the `-b` command can be used to display the interrupt request line (IRQ) addresses as seen by the individual cards.

```
$ lspci -b
00:00.0 Host bridge: Intel Corporation 82855PM Processor to I/O
Controller (rev 03)
00:01.0 PCI bridge: Intel Corporation 82855PM Processor to AGP
Controller (rev 03)
00:1d.0 USB Controller: Intel Corporation 82801DB/DBL/DBM (ICH4/ICH4-L/ICH4-M)
USB UHCI Controller #1 (rev 01)
00:1d.1 USB Controller: Intel Corporation 82801DB/DBL/DBM (ICH4/ICH4-L/ICH4-M)
USB UHCI Controller #2 (rev 01)
```

The output can also be displayed in a tree format, which is useful when a quick overview of the devices is needed.

```
$ lspci -t
-[0000:00]-+-00.0
           +-01.0-[0000:01]----00.0
           +-1d.0
           +-1d.1
           +-1d.2
           +-1d.7
           +-1e.0-[0000:02-0a]--+-00.0
           |                    +-01.0
           |                    +-01.1
           |                    \-03.0
           +-1f.0
           +-1f.1
           +-1f.5
           \-1f.6
```

lsmod

The `lsmod` command shows all the information about loaded modules, with a format of name, size, use count, and list of referring modules. Part of the listing is shown below:

```
$ lsmod
Module                  Size  Used by
nls_iso8859_1           3768  1
nls_cp437               5432  1
vfat                    9764  1
fat                    46376  1 vfat
usb_storage            86620  1
ip6t_LOG                6212  7
xt_tcpudp               2728  2
xt_pkttype              1560  3
ipt_LOG                 5708  8
xt_limit                2056  15
binfmt_misc             7872  1
snd_pcm_oss            43300  0
```

The `lsmod` command derives the data from the /proc/modules file and displays it in a form more easily read by humans.

/sys

The /sys directory contains all the files related to the kernel, firmware, and other system-related files. There are a number of directories under /sys to ensure that it is a well-organized structure. The overall structure can be seen using the `ls` command.

```
$ ls -l /sys
total 0
drwxr-xr-x   2 root root 0 2009-05-23 10:53 block
drwxr-xr-x  16 root root 0 2009-05-23 08:47 bus
drwxr-xr-x  39 root root 0 2009-05-23 08:47 class
drwxr-xr-x   4 root root 0 2009-05-23 10:56 dev
drwxr-xr-x  10 root root 0 2009-05-23 08:47 devices
drwxr-xr-x   5 root root 0 2009-05-23 08:47 firmware
drwxr-xr-x   3 root root 0 2009-05-23 08:47 fs
drwxr-xr-x   6 root root 0 2009-05-23 08:47 kernel
drwxr-xr-x 127 root root 0 2009-05-23 10:53 module
drwxr-xr-x   2 root root 0 2009-05-23 08:47 power
```

The /sys directory is related to the /proc directory, described in the next section.

/proc

The /proc filesystem is a virtual filesystem, which facilitates communication between the Linux kernel and the user processes. The /proc filesystem contains a number of directories that can organize the data below it and virtual files. Virtual files can be read from or written to the /proc filesystem as a method of communicating with specific entities in the kernel. The virtual files can share data from the user to the kernel or vice versa. It may present information in both directions, but it is not required to do so.

There are a number of interesting files in the /proc filesystem – such as *cpuinfo*, which identifies the type and speed of the processor installed in the system; and *modules*, which identifies the currently loaded modules in the kernel. A typical listing of the directory is shown in Figure 5.1.

The listing shows a series of numbered files at the left-hand part of the screen. These are directories for a process that is running on the computer. The directory labeled 1 is for the first process initiated, which will always be the `init` process.

`modprobe` and modprobe.conf File

Normally, devices can be detected during installation or when a new device is installed in a system. However, this sometimes fails, particularly with

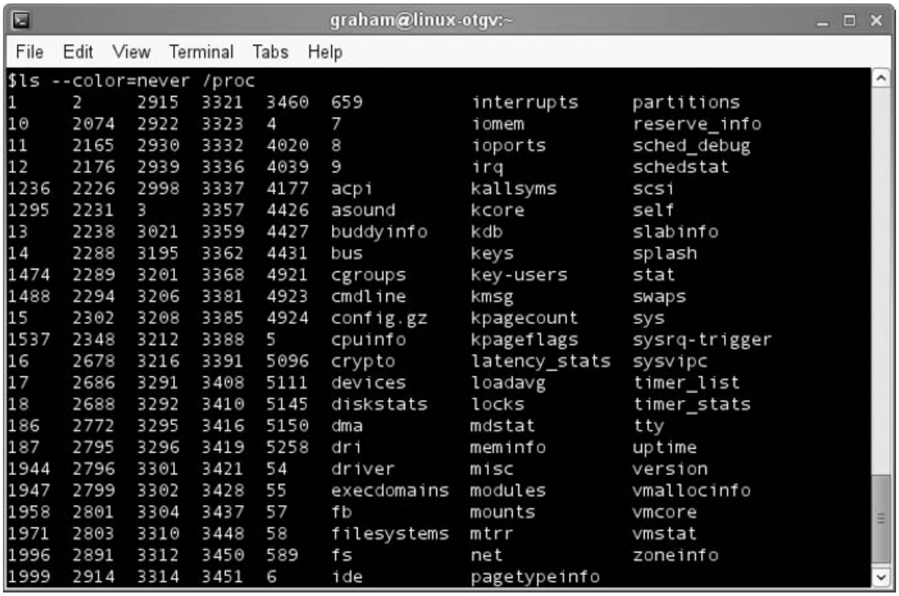


FIGURE 5.1
Example listing of the /proc directory.

hardware that is new or uncommon. When this occurs, there are a number of ways to initialize the hardware manually:

- Modprobe is the high-level handler for all modules, and it can be used to unload or load a new device's kernel module.
- The /etc/modprobe.conf.local file can be edited to prompt the system to recognize and support the new hardware upon reboot.

The base system will use the /proc/modprobe.conf file to load the modules, which should not be modified. This file will append the pathname –/etc/modprobe.conf.local to itself via an include statement. Once an entry has been added to the /etc/modprobe.conf.local file and a reboot undertaken, the system will perform a module dependency check.

As root, you can also manually load (and unload) a device's kernel module using modprobe. This command will look in the /usr/lib/[kernel version] for all the modules and files except for the optional /etc/modprobe.conf configuration file and /etc/modprobe.d directory. If the module does not exist, modprobe will generate an error. As modprobe does not do anything to the module itself, all dependancies and the resolving of symbols is handled by the kernel itself. Kernel messages generated by a module failure will have to be displayed using the `dmesg` command.

> **Note**
> The version of the kernel can be displayed using the command `uname -r`.

Each module may need one of more additional modules loaded to enable it to function correctly, and modprobe will check for these dependancies in the modules.dep file, which is itself generated by the command `depmod`. The modules.dep file is located in the /lib/modules/'uname directory.

EXERCISE 5.2: Removing a Module from the Linux Kernel

In this exercise, you will be removing a module, an Institute of Electrical and Electronics Engineers (IEEE) 1394 card, from the Linux kernel:

1. Remove the hardware from the Linux system and boot the system.
2. Once logged in, you need to remove the module from the kernel. This is done using the modprobe command, `modprobe -r ieee1394`
3. Check for any kernel messages related to the removal using `dmesg`
4. Reboot the system again, and the hardware and module should now be removed. ■

/etc/modules.conf Configuration File

The behaviors of modprobe can be altered by the optional /etc/modules.conf file. This file consists of a set of lines, which looks similar to a shell script. The file will exist only if you are installing kernel modules, which are not compiled directly into the kernel and not handled by modprobe elsewhere.

Linux Hardware Compatibility List

The devices that are supported by Linux are vast, and due to the open-source community this list is growing on a daily basis. However, every version of Linux (Debian, RedHat, and so forth) and every release of Linux will have a different set of supported hardware. Obviously, the newer the kernel, the more likely it will support the hardware device you are trying to install. Some devices will be detected and installed automatically, while others may need to be added manually.

There are numerous Web sites that detail the compatibility of systems and individual pieces of hardware, such as www.linux-drivers.org. If you are building your own Linux system or installing Linux on an existing system, it is worth examining these sites to ascertain what problems (if any) you are likely to encounter during the build. Alternatively, using the `livecd` option of a particular build can give you a very quick assessment of your hardware. The following sections will delve into the hardware support in more depth and how to load modules directly into the system.

NETWORKING

It is very rare that a modern computer system is not connected to some form of network – even if that is just a modem to access the Internet. This will therefore require the setup of a network card and TCP/IP on your system. The importance of the network card and physical connection must be understood, especially in the corporate environment. It is not worth purchasing switches and networking to run at 1 GB/s and then install 10 MB/s NICs. Where possible, it is preferable to use the same NIC when setting up a network to reduce the support overheads and allow for a smaller inventory of spares.

Configuring the Interface

When initially undertaking the network configuration, the basic rules that apply for any other device apply. If you are installing an NIC, ensure that it is on the supported device list for the Linux kernel you are installing. Most NICs

will have one or more lights to indicate whether it is working and connected correctly. If these indicate an error, use any diagnostics tools that are available to check out the device. Often, these diagnostics tools are Microsoft Windows based, so you may need to move the card to another system if the system you have does not dual boot. The reliability of devices is very good; however, it is not unknown for a new card to be faulty.

When you have installed the NIC and are confident that it is working, the NIC needs to be configured. There are numerous GUI-based tools that are bundled with Linux, and which one that is installed on your system will depend on the flavor of Linux in use and the GUI installed (KDE, GNOME, and so forth). Under newer versions of SUSE Linux, there is an application called `networkmanager`, which is available from the panel. For the purposes of this chapter, we will discuss the command-line setup of the network cards. These commands will work across all versions of Linux and will ensure that you understand the basics of the network setup.

There are two key commands that are required when configuring an NIC: `ifconfig` and `ifup`. The `ifconfig` command is very useful in displaying the status of an NIC, as shown below. It should be noted that superuser privileges will be required when using some of the options in the `ifconfig`, `ifup`, and `ifdown` commands.

```
$ /sbin/ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0D:56:E7:9D:B1
          inet addr:192.168.1.38  Bcast:192.168.1.255
Mask:255.255.255.0
          inet6 addr: fe80::20d:56ff:fee7:9db1/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:8773 errors:5 dropped:0 overruns:0 frame:5
          TX packets:2722 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:5185021 (4.9 Mb)  TX bytes:226856 (221.5 Kb)
          Interrupt:11

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:200 errors:0 dropped:0 overruns:0 frame:0
          TX packets:200 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:14324 (13.9 Kb)  TX bytes:14324 (13.9 Kb)
```

The example above shows a fully setup NIC with an IP address, transmitting and receiving packets. The following will assume that the network card

is not working and will step through the actions needed. This is especially important where multiple network cards are installed. With multiple NICs, the cards are sequentially numbered eth0, eth1, and so forth, and you must be sure that you are manipulating the correct device.

The NIC must be turned on for the system to recognize it, and you can achieve this using the command:

```
ifconfig eth1 up
```

or alternatively:

```
ifup eth1
```

Reconfiguring the IP and subnet mask of the device may require the NIC to be inactive or to be turned on and off. This can be achieved using `ifconfig` or `ifdown`, as such:

```
ifconfig eth1 down
ifdown eth1
```

Exam Warning

Remember that `ifconfig`, `ifup`, and `ifdown` are usually located in the /sbin directory, which is not part of the path as defined in `$PATH` for most users. The full directory path may therefore be required to execute the commands.

Some older NICs require the card to be set to specific IRQ and I/O addresses, and these can be set by the `ifconfig` command as well. Most modern NICs will be autodetected by the kernel, and hence this step will not be required.

```
ifconfig eth1 irq 12 io_addr 0x300
```

Once these commands are undertaken, the execution of the `ifconfig` command with no parameters should display the newly set up NIC. The IP address associated with the NIC can be assigned dynamically from a DHCP server (more details are given later in the chapter in the section "DHCP Set Up and Configuration"), or it can have a static address assigned to it. It is more common to assign a static address to servers, particularly if they are Internet-facing. To set an NIC to a specific IP address, the command is

```
ifconfig eth1 10.10.10.3 netmask 255.255.255.0
```

The `ifconfig` command should be run again to display the status of the NIC and to ensure that the IP address has been set correctly.

DHCP Setup and Configuration

For the majority of devices, the use of DHCP will be the preferred option. For small networks, these addresses are often obtained from the broadband router or similar device. For larger corporations, the use of DHCP is preferred to reduce the support and maintenance overhead of using static addresses.

The host running DHCP obtains an IP address by contacting a central server, which maintains the addresses for one or more subnets. Once the client has contacted the server, it will be assigned an IP address, subnet mask, and potentially other information such as the default route and IP address of the default name server. The information that is requested is listed in the configuration file /etc/dhclient.conf and is read upon startup, and will attempt to configure all the interfaces that are present on the system. The default is to require only the IP address configuration, although the domain name servers (DNSes) can also be required if the appropriate line is defined in the dhclient file.

A command-line DHCP client can also be used to configure the network interface, called `dhcpcd`, which is located in the /etc directory. The DNS information obtained from the server will be written to the resolvconf or, if unavailable, directly to /etc/resolv.conf. In larger networks, the DHCP server may also supply configuration data for NIS and Network Time Protocol (NTP), and these will be stored in /etc/yp.conf and /etc/ntpd.conf. These services will also be stopped and started by `dhcpcd` to ensure they are notified of the change.

For support reason, a number of other options are useful with the `dhcpcd` command:

- `--release` releases the current lease from the host and deconfigures the interface.
- `--renew` will attempt to renew the lease. These two commands are often down sequentially to check the connection to the DHCP server.
- `--nogateway` will restrict any updation of the default route.
- `--nontp` will stop the ntp being updated, /etc/ntp.conf
- `--nodns` is used to stop the DNS information being updated in /etc/resolvconf

A DHCP alternative client is dhclient, which runs on startup and reads the */etc/dhclient.conf* file to read its configuration instructions. This file will contain a list of all the network interfaces that are configured in the current system. These will, if necessary, be configured with the dhclient. If the interface is configured using this client, it stores the DHCP lease in the

/var/lib/dhcp/dhclient.leases configuration file. As the number of leases may be large, this file is recreated from time-to-time to ensure a manageable size.

Configuring a Wireless Interface

The use of wireless network cards has grown rapidly over the last few years due to the increase in speed of the wireless infrastructures. The configuration of a wireless interface is very similar to that of a wired interface. To display the status of all the wireless interfaces on a system, `iwconfig` is used with no parameters or the `iwlist` command, which is found in /usr/sbin. The wireless statistics are obtained from the /proc/net/wireless file.

To set up a wireless NIC, the main parameters to be used are given below:

- `ESSID` is used to set the network name to enable the user to roam across a number of access points.
- `mode` will need to be set depending on the network topology. The common parameters are `managed` for a network of many access points, ad hoc for a point-to-point connection, and `master` if the device will act as access point.
- `key/enc` is used to set the current encryption key, and is entered in hex notation in the form XXXX-XXXX-XXXX-XXXX or as an ASCII string with the `s:` prefix.

Network Configuration Files

There are a number of network configuration files in the systems that are modified automatically or manually. The files can have comments embedded in them using the # symbol as per normal, and it is advisable to add meaningful comments if these files are modified manually.

Hosts File, /etc/hosts

The hosts file maps actual IP addresses to hostnames. For large networks, this is usually undertaken by using DNS, as the maintenance involved in keeping hosts files current is usually large. The hosts file will contain both IPv4 and IPv6 addresses. Often, it is useful to add a small number of frequently used hosts into the hosts file to reduce the load on the network and name servers.

Services, /etc/services

The services file maps port numbers to services. A port is specific to an application and is the communication endpoint used by the transport layer protocols in the IP suite. Applications will use these port numbers to ensure that the sending and receiving systems understand which application the

packet is destined for. For example, suppose a host wishes to display a Web page located on another system. The standard port for the World Wide Web Hypertext Transport Protocol (HTTP) is 80, so the source will send a packet to destination with port 80 in the IP header block. The destination will then know to route the packet to the Web server application to execute. On a typical system, it is not necessary to modify the services file after it has been initially configured during installation.

Name Switch Service, /etc/nsswitch.conf

The Name Switch Service is used to tell the system which service to use, with potentially a number of entries per service to allow for multiple "databases" and their lookup order to be specified. Typically, there will be entries for passwd, hosts, networks, and services – among others. The typical entries are as follows:

- `nis` – use NIS or YP
- `dns` – use the domain name service
- `files` – use the local files

Typical entries would look like

```
hosts:      files dns
netmasks:   files
netgroup:   files nis
```

Resolver File, /etc/resolv.conf

The resolv.conf file will normally be constructed automatically and will never need to be changed manually. It will contain the list of one or more nameservers, typically obtained from the DHCP service. The list of servers will be de fined such as

```
nameserver 68.87.85.98
nameserver 68.87.69.146
```

TCP/IP Ports

There are a number of common networking ports that are used frequently. The entire list is very long and is organized into a number of sections. Ports 0 through 1023 are defined as well-known ports, such as FTP. Registered ports, that is those with an Internet Assigned Numbers Authority (IANA) registration, are from 1024 through 49151. The remainder of the ports from 49152

through 65535 can be used dynamically by applications. A brief description of these follows.

Port 20 and 21 These are for FTP data and FTP control, respectively. Both of these are required to be open on a firewall to allow FTP to work correctly.

- **Port 22** This is for the remote login protocol Secure Shell (SSH) and is the preferred method of connecting to a system due to its additional security.
- **Port 23**: Telnet. This is used for accessing system remotely. It is not very secure and should only be used for local hosts.
- **Port 25** This is the Simple Mail Transfer Protocol (SMTP), which is the de facto standard for electronic mail across IP networks and is used from server to server.
- **Port 53** This is the DNS protocol, which translates names into actual IP addresses.
- **Port 80** This is used for accessing Web servers as described above.
- **Port 110** This is used by the Post Office Protocol (POP) service. The current version is POP3 and is used by local e-mail clients to retrieve mail from servers.
- **Port 123** This is the NTP, which allows clients to synchronize their time with remote time servers to ensure that all systems have a consistent time.
- **Port 143** Modern e-mail clients use the Internet Message Access Protocol (IMAP) to retrieve mail from servers. This is becoming more prevalent than POP3.
- **Port 443** This is the Hypertext Transfer Protocol Secure that combines the HTTP protocol with a cryptographic protocol, which can be used for payment transactions and other secure transmission of data from Web pages.
- **Port 631** This is the Internet Printing Protocol (IPP). Clients can use this to print to printers located remotely on the network.
- **Port 3306** This is the standard port for MySQL, the standard database used by Linux.

These ports are defined in the /etc/services file on Linux systems.

Managing Connectivity

The descriptions above have outlined the basics of network connectivity and how to install a network card, either wired or wireless, into a system. The following section will build on this knowledge and define how to manage the connectivity between systems in a network. This will involve how to perform routing around the network, how to provide additional security through the use of iptables, and how to look at troubleshooting network connectivity issues.

Routing

The setup of the IP address and associated data on each of the NICs within a system is only part of the required network configuration. The system needs to know where to route packets, and this is achieved using the `route` command, which is located in /sbin. Route will be used after the interfaces have been set up with `ifconfig` or `iwconfig`. Routing is often seen as a complex task; however, once the basic concept is grasped, it is easily understood.

Any network interface will have an IP address and a subnet mask. In human-readable notation, the IP address will be in the form 192.168.1.1 (for IPv4) or 2001:db9:0:1234:0:567:1:1 (IPv6). The subnet mask will identify how many nodes are in that network; for instance, a class *C* network will have 254 nodes. When a machine has to communicate with another machine, it will look at the destination IP address and will decide how to route these packets to it. If it is on the local network, it can do so directly. Otherwise, it has to use an intermediate router to send the packets. The system will decide on which network interface to send the packet out on if there are multiple NICs in the system.

The `route` command, located in /sbin, can be used with no parameters to display the current routing table, as shown below:

```
syngress> /sbin/route
Kernel IP routing table
Destination  Gateway      Genmask        Flags Metric Ref Use Iface
192.168.1.0  *            255.255.255.0  U     1      0   0   eth0
loopback     *            255.0.0.0      U     0      0   0   lo
default      192.168.1.1  0.0.0.0        UG    0      0   0   eth0
```

The first column shows the destination IP or the hostname if it is defined in /etc/hosts of the receiving host. The default gateway for this machine is the default entry, and will be where packets are sent if no specific route exists for a destination that is trying to be reached. The genmask column defines the netmask for that particular network. The flags column can have a number of options, with `U` being the route is enabled and `G` specifying that the

destination requires a gateway. The other notable column is the Iface, which specifies which interface is used for that route.

The `route` command can add to the routing tables, located in /proc/net/ipv6_route and /proc/net/route. The command can specify a host or a network as a destination, with the default being a host if no option is used. The most common route to add is that of the default gateway such as

```
/sbin/route add default gw 192.168.1.1
```

The IP address could be substituted for a hostname if one was defined in the local hosts file. If the interface has just been configured using the `ifconfig` command, the network may have to be added by hand.

```
/sbin/route add -net 192.168.1.0 netmask 255.255.25.50 dev eth1
```

Direct, point-to-point connections can also be configured, which is useful if you have two computers (one of which can connect to the Internet through a modem). The second computer can be used as a gateway by adding in the following route:

```
/sbin/route add -host 192.168.10.45 gw 192.168.1.1
```

ipchains

Linux, like all operating systems, is vulnerable to attackers, either deliberate or accidental. Any system needs to be secured so that remote access to the system can be controlled or denied totally. This can be achieved using the Linux ipchains, which is effectively a firewall. When this is configured correctly, ipchains examines the header of the packets entering or leaving the system and deny or allow those as needed. For instance, you may want to allow your browser to view a Web page but to block the stream of advertisements that are associated with it. You can allow protocols such as Telnet out, but not allow anyone from the outside to your system.

The configuration files for ipchains are located in a set of rules in /etc/ipchains.rules. There are three sets of rules, which are chained together (hence the term `ipchains`). These three chains are the input, output, and forward. Packets entering a system are first examined by the input chain. If this chain accepts it as valid or allowed traffic, the destination of the packet is looked at to see if it needs to be routed. If it does, then the forward chain is called upon to examine the packet for validation. The output chain is consulted for the final check. The more the checks or rules in each section, the longer it takes to validate the packet and the more CPU processing that is required.

A user can set up new rules using the `ipchains` command such as

```
ipchains -1A input -1i eth0 -1s 192.168.1.0/24 -1j ACCEPT
```

This command accepts packets entering the system on eth0 from the source network 192.168.1.0/24. The command can be more specific in terms of ports and IP addresses. For instance, to block (deny) all TCP ports from 1 to 1023 leaving the system, the command is

```
ipchains -1A output -1p tcp -1 -1sport 1:1023 -1j DENY -1l
```

Alternatively, you can also block or allow UDP ports as well such as

```
ipchains -1A output -1p udp -1j DENY -1l
```

As you can see, `ipchains` is very powerful but requires careful setup. Often, applications may use more than one port, and may use one port into a system and another one out. Careful testing of `ipchains` is therefore required, and documentation of the rules you install is particularly important. Why you blocked or allowed a specific port may be forgotten months after the event.

iptables

The ipchains program was implemented into Linux from early on and is still in use in early Linux kernels. The Linux kernels 2.4 and 2.6 and later use the iptables program. Like ipchains, iptables is a user space program primarily for system administrators. It is used to configure the tables, rules, and filters to control the treatment of network packets into and out of the system. It uses the Xtables framework, which itself is used by Netfilter. In the majority of Linux systems, the program is installed as `/usr/sbin/iptables`. Xtables is the kernel-level component and provides an application program interface (API) for kernel-level extensions. The tables are associated with a number of specific kinds of packet processing. Packets are processed by the system by rules in a chain, with each rule able to send the packet to another rule, if necessary. All network packets into and out of the system must traverse at least one chain.

There are three predefined chains for input, output, and forward in the table. A packet traverses the chain until a rule matches the packet and decides what to do with it (such as accept or drop a packet), return the rule processing to the calling chain, or until the end of the chain is reached. The current rules can be displayed using the `iptables -L command`. The `iptables` command must be run as root, as it requires elevated privileges. As network address translation (NAT) is configured from the packet filter ruleset, this is included with `iptables`.

The `iptables` package includes tables for IPv4 and also IPv6, the later being designated with the table `ip6tables`. There are also a number of additional programs that work with `iptables` – the most common being `netfilter`, whose development team were primarily responsible for the

creation of `iptables`. With `netfilter` and `iptables`, a user can build stateless and stateful packet-filtering firewalls.

DNS Record Type and DNS Resolution

The DNS is used to convert names such as www.redhat.com to their actual IP address. This is primarily used, as humans can remember these far better than dotted quad IP addresses. Whether you are using a single computer hooked up to the Internet via a modem or on a large network, your system will have one or more DNSes listed. These are queried to perform this name-resolution process by the resolver process. These servers are defined in the /etc/resolv.conf file defined above. Each nameserver is part of a tree structure and will have an authoritative nameserver for its domain, such as http://foo.com. Each nameserver may delegate parts of its zone to other nameservers for convenience and speed. Starting from the root domain, the server that is the authoritative nameserver for a domain can be found by following the chain of delegations. While this may sound complicated, most users do not need to know the details of this (as the task of finding the information is handled by resolvers).

Each DNS will hold the data for the domain in resource records. These records hold a single fact about that domain, with the common records defined below:

- `A` (address) records define the actual IP address associated with a name.
- `AAAA` (IPv6) is an address record.
- `NS` (nameserver) records define the authoritative nameserver for the domain.
- `MX` (mail exchanger) records define the main server for the zone.
- `PTR` (pointer) records define the real name of the host for a particular IP.
- `CNAME` (canonical name) is the alias of one name to another.
- `TXT` (text) is primarily for human-readable text but can also contain machine-readable data.

The setting up of a DNS is covered in depth in Chapter 9 – "Installing, Configuring as a Server." The DHCP daemon that runs if DHCP is enabled on the local computer is `dhcpcd`, and this daemon will continue to run until the machine is shutdown. This will continue to try to renew the IP address lease from the DHCP server every 3 h. The messages from this server will be stored in the syslog file (usually this is either */var/adm/syslog* or */var/log/syslog*).

Network Connectivity Troubleshooting

The basics of setting up a network have been described above. The following will guide the user through basic network connectivity troubleshooting. Once the NIC has been configured and the routes added, the machine should be able to connect to a network. This may be to obtain a DHCP lease from a DHCP server if a static IP has not been assigned. When connectivity issues arise, a systematic approach is needed to ensure a quick resolution.

If the machine is newly built, it is advisable to use a network connection, which is known to be fully working. This will ensure that the physical connections, cable, and upstream devices such as routers, switches, and DHCP servers are fully operational. Often, a perceived configuration fault is tracked down to faulty network device or cable external to the machine trying to be connected.

The NIC configuration should be checked out and then connected to the network cable checked out as described above. The machine should now be initialized with an IP address and relevant nameserver information, either from a DHCP server or manually edited if a static IP address scheme is being used. The `ifconfig` command, with no parameters, should be executed to display the status of the NICs. If a DHCP server is in use and the machine successfully negotiated with it, the command would display the IP address in the output. For a more comprehensive output than `ifconfig`, the `netstat` command can be used. The output is listed by sockets (application-to-application connections between two computers). The common options for the `netstat` command are shown in Table 5.1, common `netstat` options.

Table 5.1 Common `netstat` options

Option	Output
`-a`	Shows the state of all sockets, and routing table entries
`-g`	Displays the multicast groups configured
`-i`	Shows all the interfaces configured `ifconfig`
`-v`	Verbose output
`-s`	Summary of activity for each protocol
`-c`	Output displayed every second. This is very useful in testing.
`-e`	Verbose output for active connections only
`-C`	Displays information from the route cache

It is often useful to have the `netstat` command running in a separate terminal window with the `-c` command while testing is being undertaken. Additionally, there will be an entry in the Address Resolution Protocol (ARP) table, located in /proc/net/arp. The primary use for ARP is to translate IP addresses to Ethernet MAC addresses or the actual hardware address embedded in every NIC. With the advent of IPv6, the functionality of ARP is now provided by the Neighbor Discovery Protocol (NDP).

```
$ /sbin/arp
Address             HWtype  HWaddress           Flags Mask
Iface
192.168.1.1         ether   00:12:1e:bb:3c:d2   C
eth0
```

With the machine on the network, the connections to various systems and networks can be tested. Initially, a test should be carried out to another local machine to see whether a machine on the same subnet can be seen. The `ping` command can be used to send out an Internet Control Message Protocol ECHO_REQUEST datagram to elicit an ICM ECHO_RESPONSE from a host or gateway. To ensure that your system is fully configured, you can use the `ping` command to ping its loopback address using the command `ping localhost` or `ping 127.0.0.1`.

A sample output is shown below, where the `ping` command uses the `-c` option to limit the number of pings to 3.

```
$ ping -c 3 127.0.0.1
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.065 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.066 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.064 ms

--- 127.0.0.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2000ms
rtt min/avg/max/mdev = 0.064/0.065/0.066/0.000 ms
```

In addition, the command `hostname` can be used to display the local hostname. This command can be used to display the local host name and the local IP address(es) of the host, depending on the parameters used. This will further clarify whether the local IP addressing is working and is set up correctly.

When the local machine is known to be working correctly, the command can be used to test other machines. Always start by pinging machines close to you to build up confidence in the network and supporting infrastructure. As your confidence grows, you can then start trying to ping devices on remote networks, such as the Internet. It should be noted that a lot of servers

connected to the Internet will have stopped ICMP packets at the border firewall or router to reduce traffic, so it is advisable to use other commands to test the connectivity.

The `ping` command can be used directly with the IP address, or it will do address translation using the hosts file or via resolution in one of the nameservers. If you use the notation `ping actual_name`, the command will echo back the name and, if resolved, the IP address; otherwise an error message of "`ping: unknown host`" is displayed. If you use a name of a well-known server, for example, http://ping linux.com or ping www.microsoft.com, and the name is resolved, then basic nameserver resolution is working.

```
$ ping www.syngress.com
PING syngress.com (145.36.40.200) 56(84) bytes of data.
```

To find out more about the actual route, the ICMP packets take from your machine to the target, the `traceroute` command can be used. The `traceroute` command is located in /usr/sbin, and this path may not be present in your `$PATH` statement. This listing may be long, depending on your location and the server you are trying to see. The first part of the output to the Syngress Web server is shown below:

```
$ /usr/sbin/traceroute www.syngress.com
traceroute to www.syngress.com (145.36.40.200), 30 hops max, 40 byte
packets using UDP
 1  192.168.1.1 (192.168.1.1)  1.117 ms   0.595 ms   0.621 ms
 2  * * *
 3  ge-3-27-ur02.grant.tx.houston.comcast.net (68.85.250.25)  7.015 ms
7.898 ms   7.332 ms
 4  te-8-1-ar01.royalton.tx.houston.comcast.net (68.85.244.101)  10.504
ms   10.304 ms   9.740 ms
 5  po-11-ar02.royalton.tx.houston.comcast.net (68.85.244.98)  11.640
ms   11.836 ms   11.808 ms
 6  po-17-ar02.greenspoint.tx.houston.comcast.net (68.85.244.130)
13.299 ms   13.271 ms   13.276 ms
 7  te-0-1-0-4-cr01.dallas.tx.ibone.comcast.net (68.86.91.57)  17.153
ms   17.074 ms   16.860 ms
 8  64.132.69.249 (64.132.69.249)  16.837 ms   16.650 ms   16.243 ms
 9  64-132-52-114.static.twtelecom.net (64.132.52.114)  61.314 ms
61.206 ms59.852 ms
```

Along a network, there will be a number of routers, which interconnect different networks together. These routers can also filter the traffic and act as firewall. This often means that the `traceroute` command may display an "*" to show it did not receive a response, but the next hop may show a valid response. Often these can be used to determine where a firewall is located

in the network. The routers along the network decide where (if anywhere) to send the packet; that is, to forward it to one of its interfaces. If the router does not find a matching route for the packet, it will be sent to its default route (and so on) until the packet reaches its destination. The router has a routing table in basically the same format as we saw earlier for the Linux machine, although it will be set up in a different manner (unless this is a Linux machine with two or more network cards acting as a router!) Routers come in all shapes and sizes, from the very small to large and extremely expensive devices located in major network switching centers.

In a local network, the routers are often managed by using the `telnet` command. Recently, the use of telnet has waned, as there are a number of security issues with the protocol (namely, no encryption involved, so passwords and other data is sent in the clear text), and the use of SSH is much more prevalent. Telnet is useful as the port it connects on can be specified, so the command can be used to log on to many different hosts. This may be particularly useful if you need to add a route to a local router that you administer to ensure the connectivity is correct. It can also be used to display part of a Web page, for example, or at least the initial Hypertext Markup Language (HTML) from that page. This will also mean that a Linux system installed on a very small system with few applications, and perhaps no GUI, can still be used to test the routing.

Learn by Example: Testing Network Connectivity

Having trouble with the connectivity between my system and a server on the system, I systematically diagnosed the problem. First I checked whether the NIC was correctly installed and had the routing table setup, and that the system was connected to a network using the appropriate network cable. This was achieved by using the `ping` command to test the *loopback interface*. Once satisfied this was all working, I used the `ping` command again, but using the remote servers IP address. Upon correctly responding, I used the remote servers name instead of IP address. Once all these tasks are complete and working, it can be assumed that the TCP/IP network card and routing is working correctly.

Earlier in this section, you learned that when a system does not seem to recognize a name but works perfectly with IP addresses, then there is an issue with the name resolution. First, if possible, discover if it is a global issue with all your machines on the network by utilizing the `ping <server name>` command on another machine. If this works, then the problem is with your machine setup. If you do not have any other machines to test this on, try a few different names – and make sure you type the name in correctly! Check

that there is a valid nameserver defined in the /etc/resolv.conf file and that you can traceroute to that server. If you cannot perform basic routing to these servers, then name resolution will not occur.

The Domain Information Groper or `dig` command can query the nameservers listed in the /etc/resolv.conf file, and then undertakes an NS (nameserver) query. An example of the `dig` command and output is shown below:

```
$ dig syngress.com

; <<>> DiG 9.5.0-P2 <<>> syngress.com
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 54845
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;syngress.com.                  IN      A

;; ANSWER SECTION:
syngress.com.           300     IN      A       145.36.40.200

;; Query time: 207 msec
;; SERVER: 68.87.85.98#53(68.87.85.98)
;; WHEN: Thu May 14 14:58:26 2009
;; MSG SIZE  rcvd: 46
```

It can also perform a reverse lookup, where the IP address is used instead of the name. This produces slightly different results.

```
$ dig 145.36.40.200

; <<>> DiG 9.5.0-P2 <<>> 145.36.40.200
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 1949
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 0

;; QUESTION SECTION:
;145.36.40.200.                 IN      A

;; AUTHORITY SECTION:
.                       900     IN      SOA     A.ROOT-SERVERS.NET.
NSTLD.VERISIGN-GRS.COM. 2009051401 1800 900 604800 86400
;; Query time: 144 msec
```

```
;; SERVER: 68.87.85.98#53(68.87.85.98)
;; WHEN: Thu May 14 14:59:37 2009
;; MSG SIZE  rcvd: 106
```

The slightly outdated command `nslookup` (superseded by the `dig` command) is still useful, in both interactive and noninteractive modes. `nslookup` enters interactive mode when no arguments are passed to it, as shown below.

```
$ nslookup
> www.syngress.com
Server:         68.87.85.98
Address:        68.87.85.98#53

Non-authoritative answer:
www.syngress.com        canonical name = syngress.com.
Name:   syngress.com
Address: 145.36.40.200
```

EXERCISE 5.3: Troubleshooting IP Address Conflicts

On occasion, there are address conflicts within a system. To resolve this, you can do the following.

1. On the system (or system) that appears to have the IP address conflict, use the `/sbin/ifconfig` command to check the network cards and to look at the hardware and IP addresses.
2. On the DHCP server, check the ARP addresses associated with the IP address and see which system it matches.
3. Release and renew the IP address locally using the `dhcpcd` command

   ```
   dhcpcd --release
   dhcpcd --renew
   ```

4. Check the IP address. It should now be a unique address. If there are still conflicts, a static address may have been assigned to another system. In this case, turn off your system and use traceroute from the DHCPO server to locate the rogue system. ■

Remote Access

There are a number of methods to access a Linux system from another system, either on the local area network (LAN) or externally (often via the Internet). These accesses can be to simply retrieve data, using commands such as via a Web browser or to run a shell on the remote machine, using

commands such as `telnet` or `ssh`. As many of these commands have been implemented on different operating systems, the two ends of these connections may be completely different.

Telnet

To easily connect to a Linux system, the `telnet` command can be used. When the two computers are connected using the `telnet` command, you are effectively accessing the remote computer as if you were sitting directly on it using a terminal session. The terminal client may need to be set to an appropriate terminal emulation; and typically, a standard emulator is vt100. The session can be set as such using `TERM=vt100`.

As discussed in the user profiles section of this chapter, this is an important environmental variable. In addition, if the remote terminal is being accessed through a firewall, the appropriate port needs to open. The telnet port is 23, and a rule allowing this from the source IP address to the destination needs to be set.

File Transfer Protocol and Trivial File Transfer Program

The File Transfer Protocol (FTP) and trivial file transfer program (TFTP) can be used to upload and download data to a remote system. The FTP client connects to an FTP server on the remote computer, with the possibility that each computer was running a different operating system. The TFTP protocol does not allow any form of security and is generally not used except on computers that require a separate level of authentication. It is often used in routers as a simple method of transferring configuration data to and from the device. Access to a Linux server using TFTP is highly discouraged due to the insecure nature.

Secure setup of remote access solution to a Linux server is discussed in-depth in Chapter 10, "Securing Linux," and is a major focus of the Linux+ exam.

SUMMARY OF EXAM OBJECTIVES

In this chapter, you learned about how to configure the base Linux system. The basics of user and system environment variables and how to configure these globally and for individual users were explained. The main environment variables that are commonly defined were explained.

The management of devices is very important and is often complex to many individuals initially. How to add and delete modules to the kernel

and where the configuration files are located is essential knowledge for any system administrators. These commands are worth experimenting with, and learning more about them to ensure you can master them.

The majority of systems are networked together, and this chapter explored the basics of networking. The routing tables, and how to manipulate these, were explained. In modern systems, incorrect routing often causes problems. In addition, how to set up NICs with static and DHCP IP addresses was discussed, along with an introduction to DNS.

SELF TEST

1. Your manager wants you to change the system prompt for users on their system to reflect that the company has merged with another company and has rebranded itself as Plix. The manger wants user to have the prompt as "plix>" (without the " marks). What is the correct method to undertake this?

- **A.** Put the following in the /usr/.bashrc file `PS2="plix> "`
- **B.** Put the following in the /usr/.profile file `PS1="plix> "`
- **C.** Add the following to the ~/.bashrc file for each user `PS1="plix> "`
- **D.** Insert the following into the /etc/env file `PS1="plix> "`

2. You wish to run a program called disp_rights you have developed and placed in your home directory. You have limited its rights so that it can only be executed by yourself. Your username is syngress. Which command will execute the program?

- **A.** `syngress/disp_rights`
- **B.** `~/disp_rights`
- **C.** `~/syngress/disp_rights`
- **D.** `/usr/home/syngress/disp_rights`

3. You wish to find the default mail server for the domain mycorp.com. Using the `dig` command, you display the information currently held in the DNS. Which of the resource record types below correctly defines the mail server's IP address?

- **A.** NS
- **B.** A
- **C.** MX
- **D.** MS

4. You are testing the connectivity to a server with an IP address 10.10.10.4. You want to display a continuous output to the screen so that you can see when the remote server is up. The correct command is

   A. `ping -c 10.10.10.4`

   B. `ping -v 10.10.10.4`

   C. `ping -d 10.10.10.4`

   D. `ping 10.10.10.4`

5. There has been a new system installed on your network that you do not know about. You have performed a port scan and can see that port 80 and 22 are open. From your knowledge of the service ports, this is likely to be which of the following:

   A. a Web and FTP server

   B. a Web and SSH server

   C. a Web and Telnet server

   D. a Web and mail server

6. You need to change the IP address of an NIC assigned eth0, which has been assigned a static address to 10.10.100.45. This is installed in a server in a small-company network. Which of the following will change the IP address most effectively?

   A. Change the /etc/hosts file to reflect the new IP address and reboot the system.

   B. Change the address in the routing table and force a reboot using the command:

   `/sbin/route add 10.10.100.45 netmask 255.255.255.0 dev eth0`

   C. Use the `ifconfig` command to change the address:

   `ifconfig eth0 10.10.100.45 netmask 255.255.255.0 up`

   D. Change the IP address on the module using the following:

   `/sbin/ipchange 10.10.100.45 netmask 255.255.255.0`

7. A user has a computer that does not use DHCP and he/she cannot use hostnames in any command, although IP addresses do work. Where would you look to see how the DN server is defined?

   A. /etc/resolver.conv

   B. /etc/resolv.conf

   C. /etc/hosts

   D. /etc/sysconfig/network

8. A user wants to ensure that his wireless card installed correctly in his system only connects to his company's network. This network has an SSID of mycorp and uses WPA2 for added security. How would you ensure this occurs?

A. `iwconfig essid mycorp`

B. `iwconfig default mycorp`

C. `iwconfig default_ssid mycorp`

D. `iwconfig noroam essid mycorp`

9. You have just started work as an IT contractor in a company called mycorp. You have been given a laptop with a static address as you manage some of the routers, and these have access control lists (ACLs) on them allowing your IP to access them. You want to check that your nameserver, which is currently set on your machine to be 10.10.100.67, is correct. What command would most likely give you the correct result?

A. `traceroute 10.10.100.67`

B. `nslookup 10.10.100.67`

C. `dig dns.mycorp.com`

D. `nslookup dns.mycorp.com`

10. You are working in a medium-sized company and have added a new server to the network. A static IP address of 10.10.100.45 has been assigned to it. The routing table is shown below:

```
Destination  Gateway Genmask       Flags Metric Ref Use iface
127.0.0.0    *       255.0.0.0     U     0      0   0   lo
10.10.100.45 *       255.255.255.0 U     0      0   0   eth0
```

What command will add a default route to 10.10.100.1?

A. `/sbin/route add default 10.10.100.1`

B. `/sbin/defaultroute add 10.10.100.1`

C. `/sbin/route add default gw 10.10.100.1`

D. `/bin/route add default gw 10.10.100.1`

11. You wish to set up the default editor in your environment to be the vim editor instead of the current setting of `vi`. Which would be the best solution to achieve this?

A. Modify the */etc/env.conf* file to set the default editor environment variable `EDIT` to be `vim`

B. Modify the */etc/env.conf* file to set the default editor environment variable `EDITOR` to be `vim`

C. Change the *~/.bashrc* file to set the default editor environment variable `EDIT` to be `vim`

D. Change the */etc/.bashrc* file to set the default editor environment variable `EDIT` to be `vim`

12. Your employer wants to protect a finance server that has just been installed on their network and has installed ipchains on the system. What is the best description of the ipchains that has been installed?

A. Force external users to log in if they use telnet into the system.

B. Block all traffic into the system, apart from that defined in the */etc/ipchains/allow.conf* file.

C. Accept or deny packets based on the */etc/ipchains.rules* file.

D. Force the system to check all packets entering the system, as defined in the */etc/ipchains/ipchains.rules*

13. A user is experiencing connectivity issues with a network port that has been working successfully for a number of months. You have tested the network by connecting another laptop to the same port, which worked. Looking at the hardware, you can see that they have an old NIC and you wish to replace it with a new one. The kernel did not recognize the NIC upon reboot. How would you add the card manually?

A. `modprobe 8139too`

B. `modprobe enable 8139too`

C. `modprobe up 8139too`

D. `add_dep module 8139too`

14. Which of the following directories is the primary location for the current hardware information of your computer?

A. /sbin

B. /proc

C. /lib/modules

D. /etc

15. Which of the following configuration files are typically associated to individual user logins with the Bourne shell?

A. ~/.bashrc, /etc/profile

B. ~/.bash_profile, /etc/profile

C. ~/.bashrc, ~/profile

D. /etc/.bashrc, /etc/profile

SELF TEST QUICK ANSWER KEY

1. C
2. B
3. C
4. D
5. B
6. C
7. B
8. A
9. B
10. C
11. C
12. C
13. A
14. B
15. A

This page intentionally left blank

CHAPTER 6

Using BASH

Exam objectives in this chapter

- BASH Commands
- Scheduling Tasks
- Managing Services

UNIQUE TERMS AND DEFINITIONS

- **Bourne again shell (BASH)** The sh-compatible command line interpreter that executes commands read from the standard input or from a file. It is the default shell on most Linux distributions.
- **Command line interpreter or Command line interface (CLI)** A full-screen or windowed text-mode session where the user executes programs by typing in commands with or without parameters. The CLI displays output text from the operating system or program and provides a command prompt for user input.
- **Graphical user interface (GUI)** A design for the part of a program that interacts with the use and uses icons to represent features of programs. GUIs typically work with "mouse-able" interfaces with pull-down menus, dialog boxes, check boxes radio buttons, drop-down list boxes, scroll bars, scroll boxes, and the like.

- **Process ID (PID)** A unique number assigned when a new process (program) is started, and used to reference that process.
- **Parent process ID (PPID)** The PID of the process that spawned (started) a new process.

INTRODUCTION

Some people claim Linux is inferior due to its extensive use of the command line interface (CLI). While newer versions of Linux have made it increasingly easier to avoid using it, the power and versatility of the command line is one of the key strengths in the entire system, if you are willing to invest the time to really learn it. It reminds me of "The Force" – surrounding all things and binding them. The command line is your light saber, "an elegant weapon from a more civilized age,"[1] but formidable in the hands of a Linux Jedi. Step forward and begin your training!

Hyperbole? Maybe, but the command line has a number of strengths. It is easily scripted for repetitive tasks, which can then be scheduled and implemented across multiple machines – a key advantage for administrators of multiple systems. It is consistent across different distributions and versions, different desktops, and window managers. It is straightforward to manage through remote access (ssh), and ubiquitous – the exam assumes you will be managing a desktop or server, but Linux is showing up in countless embedded appliances, and their full power is seldom available from the graphic interface basic users are allowed to see. A commanding knowledge of the CLI is the key to passing the Linux+ exam and will distinguish you as a valuable asset in any company. Be warned, though: like any powerful tool, it can also wreak untold havoc if you don't know what you are doing.

So, what is BASH, exactly? It's a command interpreter, a way for a user to submit instructions to the computer. One of the original UNIX shells was the "Bourne" shell, invoked with an "`sh`" (short for *shell*). An updated version was developed and (a bit whimsically) called the "Bourne again shell" (BASH).

> **Note**
> Although BASH has evolved into the current standard in most Linux, BSD, and OSX (since v10.3), there are several other shells to choose from. Other popular ones are `zsh`; `ksh` ("korn" shell); and "`csh`" (sea shell), which uses syntax (and scripting) common to the "C" programming language, so it's popular with C/C++ programmers.

If you'd like to try some of the others, use your favorite package manager to install one, and invoke it from BASH by typing `zsh`, `ksh`, or `csh`. Bash is cleverly named, "`bash`." The shell setting can be made permanent per user by editing the users' /etc/passwd file. On a typical Linux installation, "`sh`" is a link to /usr/bin/bash.

BASH COMMANDS

How do you get to a command line? If you haven't installed a graphical user interface (GUI), the system will boot you directly into a command shell. Otherwise, you'll have to find your systems terminal program. In the version of openSUSE used in this book, it's called `Terminal` and located at the bottom of the list of programs that pops up when you click on the green lizard **start** button on the bottom left corner of the screen, as shown in Figure 6.1.

The terminal program is used so often that you may find it convenient to drag the Terminal icon to the taskbar for more direct access.

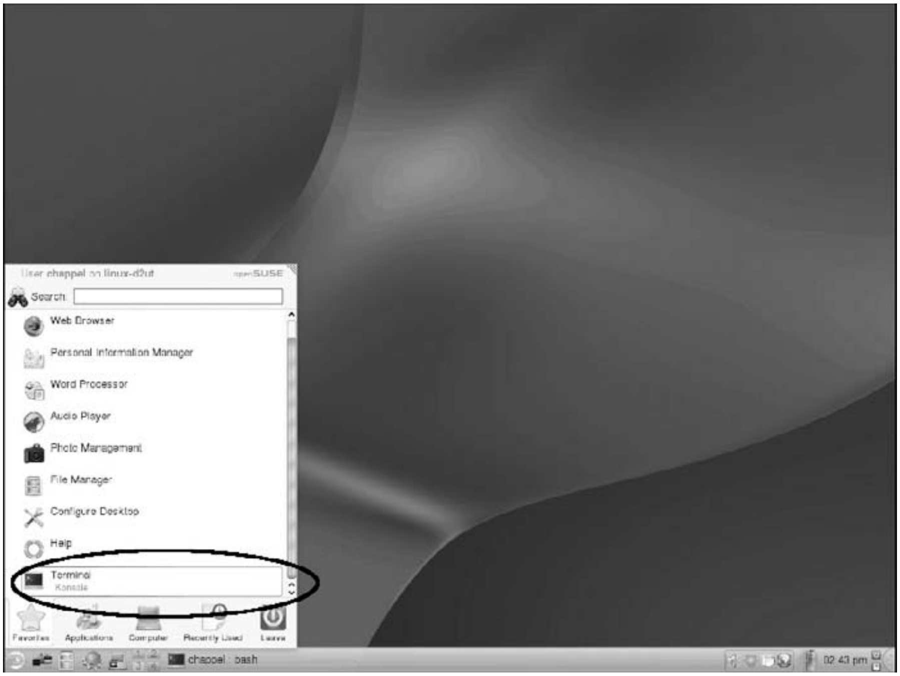


FIGURE 6.1
The terminal program `konsole` *in KDE.*

> **Note**
> You can also type **Alt + F2** to pop up the "run an application" window, and type in `konsole` in KDE or `gnome-terminal` if you use the gnome desktop.

It is frequently convenient to run several command shell windows simultaneously; you can read a "man" page describing the use of a command while trying the command, or monitor a log file while performing system management. Konsole (KDE's terminal program) supports multiple tabs with a separate instance of a command line in each, which can be convenient.

> **Note**
> To make it easier to keep track of multiple windows, `konsole` (and `gnome-terminal`) lets you rename each terminal instance – per tab or discrete window. Try it! In `konsole` type **Ctrl + Shift + N** (or click **file** | **New Tab**) to open a new tab (the tabs themselves appear on the bottom of the `konsole` window), and **Ctrl + Alt + s** (or click **edit** | **rename**) to rename it. Note that the icon for `konsole` on the taskbar at the bottom of the screen is also renamed, making it easier to go straight to the terminal window you want when you have several minimized or covered.

While digging through the various terminal program options, feel free to play with the settings for background, transparency, font, foreground/background color, and others. It isn't on the test (but hey, you're going to be staring at this for countless hours); therefore, you might as well make it something that's easy to read and pleasing to look at.

> **Exam Warning**
>
> Note that if you don't use a "monospaced" font (where every character is the same width on the screen) columns won't align correctly, and some commands use a variety of colors, making the output invisible if your background matches one of them. A pretty blue background, for example, will render all directories invisible in the default output of the `ls`' command and can induce momentary heart failure.

Navigating Directories

OK, you've got a terminal window, now what? A good starting place is basic directory navigation. Type `pwd` (*print working directory*) to see what directory you are in. `konsole` will start you in your home directory: /home/(your_user_name)/. You can see the contents of your current directory

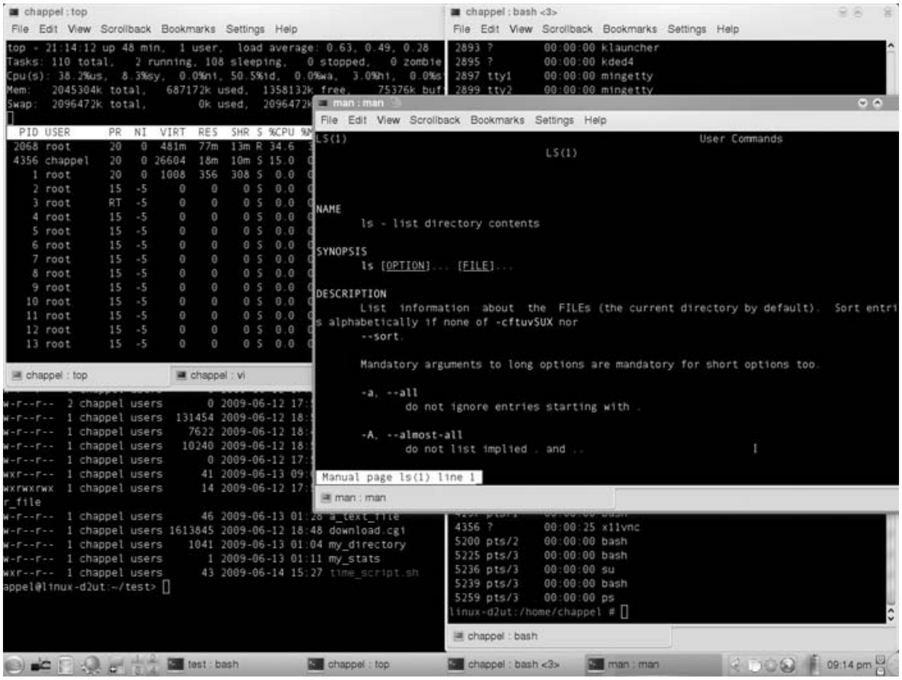


FIGURE 6.2
The KDE "`konsole`*" windows; power at your fingertips.*

by entering `ls` (to *list* things). Enter `ls` with a path to see the contents of other directories (try `ls/`, `ls/etc`), and use `cd` (*change directory*) to change your current working directory (`cd/`, `cd/etc` – try using `pwd` and `ls` in each directory as you move about).

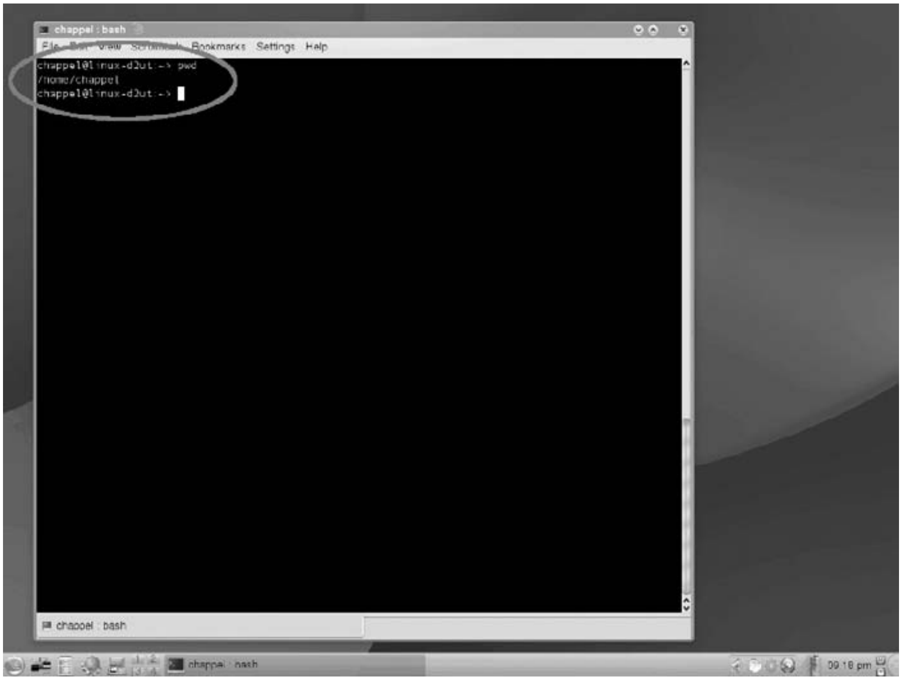
If you find yourself frequently jumping back and forth between directories, you may find it handy to use `pushd` and `popd` as an easy way to save a directory (or several). These utilities work like a stack of index cards. Think of `pushd` as writing a directory on an index card and placing it on a stack, and `popd` as picking up the card on the top of the stack and changing to that directory. Try `pushd /var/log` and notice that the current directory is now /var/log.

When you first open a terminal window, it will give you a command prompt, which is typically your machine name and username, or maybe your current directory ending with a "$" for a normal user or a "#" if you are running as "root" (the administrator account) – see Figure 6.2.

Using File Commands

The Linux file structure, like most computer systems, is a series of files, folders, and subfolders. Unlike Microsoft-based systems, there is no concept of "drive letter" – no 'C:/'. The very "top" of the files is the *root* directory – think

FIGURE 6.3
Standard command prompt.



of the files and folders as the roots of a tree. The root folder (called "/") is at the top, with folders and subfolders branching downward from there (that is, "/home/chappel/Documents" for the Documents folder in user chappel's home directory).

So, if you open a command window while logged in as user chappel, you'll see a prompt and be in the present working directory, as shown in Figure 6.3.

Clearly, manipulating the files and folders on the system is an important capability. Although GUI file managers make it easy to do many basic filing tasks, they can't match the power and flexibility at the fingertips of a command-line master.

Starting with the basics, `mkdir` creates a directory and `rmdir` deletes it (provided there are no files in the directory). The command `touch` will create an empty file; `cp` will copy the file, `mv` will move (or rename) it, and `rm` will delete it.

> **Note**
> If `touch` seems like an odd name for a command that creates files, it's because it was originally intended to change the timestamp on an existing file – so you could see the last time it had been "touched," literally. This is very handy for keeping track

of the most current version of a file, or comparing files for archiving. The fact it will create a file if one doesn't already exist makes it convenient for playing with test files.

Try it out! If you aren't already there, go to your home directory (you can simply type `cd` without any parameters and it will change to your home directory) and go through the following exercise to make some files and directories.

EXERCISE 6.1: Using File and Folder Tools

In this exercise, we'll try some of the commands for managing files and directories. Type the following:

1. `pwd` to confirm you are in your home directory (that is, "/home/<your_user_name>")
2. `ls` to list what files are there to start with
3. `mkdir testdir` to make a new directory
4. `ls` again to see your new directory
5. `cd testdir` to change to your new directory
6. `pwd` to confirm you are really there
7. `touch testfile1` to make a new file in your new directory
8. `ls` to list your new file
9. `copy testfile1 testfile2` to make a copy of your new file
10. `ls` to see both the files
11. `mv testfile2 testfile3` to rename the file
12. `ls` to show the new filename
13. `rm testfile3` to delete the file
14. `ls` to show it's really gone. ■

There you have a quick and easy introduction to file commands, but hardly "unleashing the power." To do that, we need to discuss command options and parameters. The commands you type into the command line are programs, and what they do can be controlled based on options you give.

For example, you've been using the `ls` command to list the contents of a directory. By default – if not instructed otherwise – the `ls` command will show the contents of only the current directory, limited to just file and directory names, with color-coded file types, formatted in rows, sorted alphabetically. Each of those options can be changed by passing the appropriate option flag. The general format of a BASH command is

```
command -option(s) parameter(s)
```

For the `ls` command that looks like this:

```
ls -l /etc
```

Using the `l` option (as in "long") shows not just the file names, but lots of other interesting information, such as the size, owner, creation or last modified date, and access information. Options can be stacked, so showing the same information sorted by date looks like this:

```
ls -l -S /etc
```

or you can run the options together after a single "`-`" like this:

```
ls -lS /etc
```

Exam Warning

Pay careful attention to capitalization. Unlike the various Microsoft products, all flavors of UNIX (Linux, BSD, and even OSX) distinguish between upper and lower case, in commands, options, and file names. In the preceding example, `ls -s` shows file sizes, where `ls -S` sorts by size. Likewise, FILE1, File1, and file1 are three different files; and `ls` is a valid command, but "LS" is not.

You can even list multiple directories by using the recursive option (`-R`) and/or listing multiple directories as parameters:

```
ls -R /etc /home
```

Most commands have so many options that there aren't enough letters to assign, so you can often use a long-hand notation, which is a double-dash (--) and an entire word. This requires more typing, but is often easier to remember. The command to show all the available options for `ls` uses this:

```
ls --help
```

Don't be intimidated by the long list of available options; many of them are intended to make `ls` work better within small programs called *scripts*. You

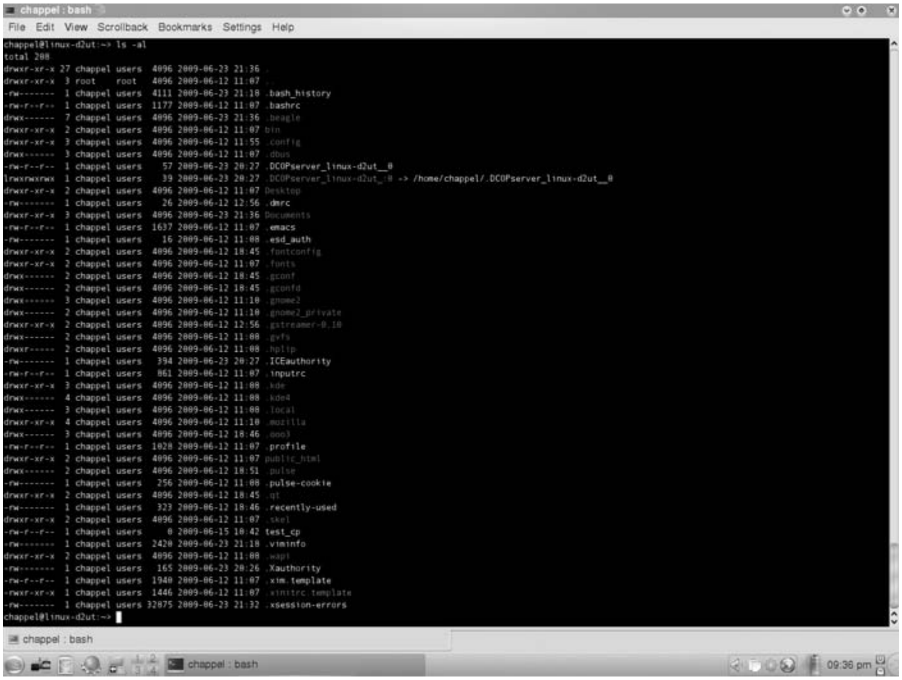


FIGURE 6.4
An example of the `ls -al` *command.*

certainly won't be expected to know all of them. Two options it is important to know are `all` and `long`:

```
ls -al
```

This shows detailed information about all the contents of a directory, even hidden files (any file that starts with a period is considered "hidden" and won't be displayed by a default `ls`), as shown in Figure 6.4.

A *d* in the first position of the first column indicates a directory; - is for a regular file. The next nine characters show the access rights or mode of the file or directory (covered in Chapter 10, "Securing Linux"). The number in the next column shows how many other files share a hard link with that file (links are covered shortly, in "File Types" section). Next is the owner and group of the file (in this case, the owner is chappel and the group is users – also covered in Chapter 10, "Securing Linux"), then the file size and the date the file was last changed, and finally the file name.

In addition to options, the parameters of a command can also alter the results it will give. The most common parameter variations are the wildcards "*" and "+." These are symbols that can represent multiple patterns; "*"

for any number of any character and "?" for just one of any character. For example, if we have a directory that contains the following:

file1
file11
file111
file2
file22
file222

`ls` (and `ls *`) will show all six files; `ls file1*` will show file1, file11, and file111 (all files starting with "file1" and followed by zero or more other characters); `ls file?` will show file1 and file2 (all files starting with "file" followed by exactly one character). Note that multiples are valid: `ls file??` shows file11 and file22. The wildcards can also be used in the middle of the parameters; to show file111, type:

```
ls f*111
```

They can also be combined; for example,

```
ls f*1?
```

Of course, wildcards can be used with any command that takes parameters; so, whereas `ls f*11` lists all files in a directory that start with "f" and end with "11," `cp f*11` copies the same list of files, and `rm f*11` deletes them.

> **Note**
> For more advanced file selection, you can use *Regular Expressions*. These define a complex method of specifying patterns to be matched and are supported in many utilities, not just in Linux. For more information, go to www.regular-expressions.info/.

Once you start digging around in the file system, you'll probably find the `ls` command a bit awkward if you are looking for something and aren't sure which directory to search in. Fortunately, there are other commands to help, including the following:

- `find`
- `locate`
- `slocate`
- `which`
- `whereis`

Of the group, `find` is the most flexible. It requires a place to start, and what to look for:

```
find /home -name file1
```

It works by doing exactly what you would do – going to the /home directory and looking for a file named "file1," then continuing to look in each subdirectory. Computers are much faster at this than their users, but if you have a large file system (or start a "/") it can still take quite a while. Providing the `-maxdepth` option will limit how many levels of directories it will search through – for example, `find /home -maxdepth 2 -name ".*"` would search for hidden files in all users home directories, but not in subdirectories. Note that the quotes aren't necessary when searching for a specific name, but are required for certain combinations of wildcards. There are no penalties for using quotes; when in doubt, leave them in.

There are a mind-boggling variety of things that can be searched for file sizes, dates, owners, permissions, and more. As with the `ls` command, many of the available options are more useful for use in scripts (for backing up files or automatically cleaning up temporary directories, for example). One particularly good use for `find` is searching for files owned by a particular user whose account you'd like to delete. This prevents files from being orphaned, or unexpectedly becoming owned by a new user who happens to get the same account ID.

```
find /home -user user_name
```

Although `find` works its way through the given directory structure, `locate` is a bit more intelligent and creates a database of file names that it searches through. This makes `locate` much faster and more efficient, but it can only find information that makes it into the database; so changes since the last automatically scheduled database update won't be found, and it can only search on file names. The `slocate` flavor of `locate` works the same way, but adds additional security to prevent users from searching through files they don't have access to. Linux systems that use `slocate` generally point the `locate` command to it, so you may actually be using `slocate` without realizing it.

Note

The default openSUSE installation doesn't include `locate`, but it can be added using the standard package manager. With the installation DVD inserted, type the following:

```
sudo zypper install findutils-locate
```

The `whereis` command is intended for searching for commands and files related to them – source, binary, configurations, and help files. It doesn't look in user directories at all.

The `which` command looks through your path (see the section entitled "System and User Profile and Environment Variables" in Chapter 5, "Configuring the Base System" for more information about the `PATH` variable) and will tell you specifically what program you will run if you type a given command. If, for example, you have a couple different compilers installed on your system, it will tell you exactly which one you'll get if you type **gcc**.

File Types

If you'll recall the `ls -al` command, you may still be wondering about all those different file types that it showed. It is jokingly said that in UNIX "everything is a file"; now we'll pull back the covers and see what that really means.

We've already worked with two types of files – regular files (`touch file1`) and directories (`mkdir testdir`). In addition to these, there are also

- hardlinks
- softlinks
- device files
- named pipes

A proper understanding of linked files requires a little technical background information. In the standard UNIX filesystems, directories contain the names of the files that reside in them, and pointers to an inode for each file. The inode stores the metadata for each file – the owner, size, access rights, time of last modification, and where the data the file contains is physically located on the actual storage device. The extra layer of abstraction provided by the inode means that it is possible to have two different names that point to exactly the same file information. This is called a "hardlink," and created using the `ln` command:

```
ln file_name hard_link_name
```

The hardlink name is just as valid as a name as the original filename, and all hard links must be deleted before the space occupied by the file is freed. You can use the `ls -l` command to show link information; the second column (between the "`-rwxrwxrwx`" and the owner name) has a number that indicates how many hard links point to that same file. The nature of hard links does not permit hard links to directories or to files residing on separate drive volumes.

> **Note**
> Although not included in the scope of the Linux+ exam, the `stat` command shows the contents of the inode for a given file, and `ls -i` shows the inode number.

In contrast to the hard link is the symbolic link, or soft link. The soft link is an actual file, with its own inode number (rather than a shared one, like a hard link). The file contains the path to the object the file links to. As a higher level of association, soft links are allowed to point to directories and to files on other file systems; but because they are independent files, it is possible to delete the target of the soft link and leave the link file "broken," pointing to a nonexistent file. The `ls -l` command shows symbolic links, with an "l" at the very first position in the row and with an arrow (->) pointing to the target file. Symbolic links are created with the same `ln` command, but with the `-s` option:

```
ln -s file_name soft_link_name
```

For more information about links, check out the Linux Information Project page at www.linfo.org/hard_link.html.

Device Files

Anything capable of moving information in or out of your Linux system has a device file in the /dev subdirectory. These files are the user accessible side of the device files, covered in "Device Management" section in Chapter 5. This way of thinking of devices makes it easy for user programs to send or receive information; they only need to write (or read) from the appropriate file. For an example of just how easy it is, try this:

```
ls -al /etc > /dev/audio
```

Make sure your speakers are turned down; it isn't exactly Beethoven.

There are a number of types of files, but the most common (and the ones covered on the exam) are the following:

- `l` symbolic link
- `d` directory
- `-` normal files

These are all file types we've already seen, plus these "special" file types:

- `c` character special file
- `b` block special file
- `p` named pipe

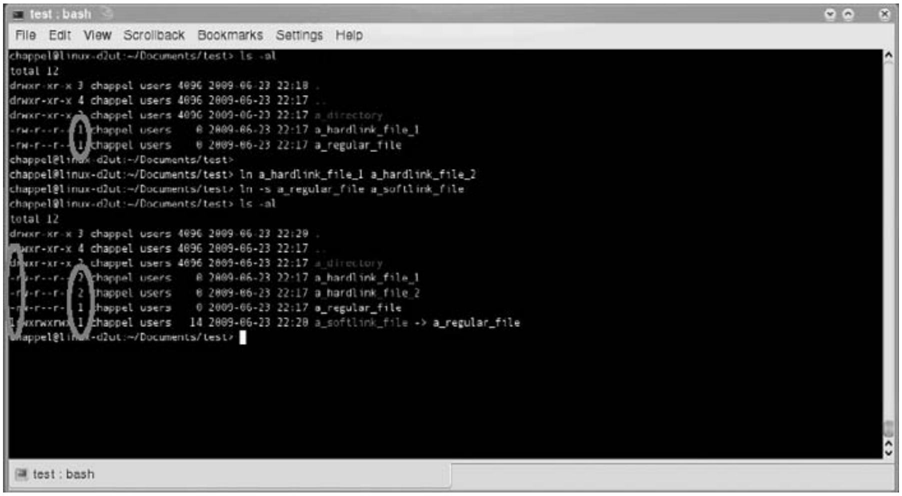


FIGURE 6.5
Output of `ls -l` *showing file types in the first column of each row.*

The first character in the above list corresponds to the first character on each line of output from the `ls -l` command, as in Figure 6.5. Notice that while symbolic links have their own file type, hard links show up as regular files, and are only distinguished by the number in the column between the access rights and the owner being greater than 1.

The block special device files are used to move data in and out of hardware connected to the system in large chunks, and use buffers to improve efficiency. Examples of block files include the systems SCSI, IDE and other hard drives, USB, CD, DVD, and tape drives (the `mount` command will show which drive corresponds to what device file). Character devices files are used to move data a single character at a time and are unbuffered; examples include modems, terminals, and other serial devices. Note that not every file in the /dev directory corresponds to a currently connected device; most are simply placeholders waiting for a new device to point to.

Named pipes first-in first-out (FIFO) are buffers that are used for communications between programs running within the computer. One program opens the pipe to write, and the other to read, and data gets transferred between them.

Special files are created using the `mknod` command:

- `mknod new_file_name c [major_device_number] [minor_device_number]` – makes character files
- `mknod new_file_name b [major_device_number] [minor_device_ number]` – makes block files
- `mknod new_pipe_name p` – makes pipe (FIFO) files

Manually creating special files normally only happens when manually installing device drivers, which should include detailed instructions with additional information.

> **Note**
> The official list of names for devices that could potentially be found in the /dev directory can be found on the Linux Assigned Names And Numbers Authority (LANANA) at www.lanana.org/docs/device-list/devices.txt.

Testing Files

Finally we come to ways to test files with the `file`, `test`, and `ls -F` commands.

Although files frequently have an extension to help indicate what they contain (.mp3 for music or .jpg for pictures), they may not be so obvious. Luckily, Linux provides the `file` command to help sort through things (see Figure 6.6). Of course *you* would never forget what type of file you made, but it can be very useful if you need to sort through user directories, or want to confirm a file is the type you expect before running it through a script.

The `file` command has a syntax, which is simply `file [filename]`, and it accepts all the standard wildcards.

The `test` command can evaluate expressions, as in `test 1 -gt 2`. It isn't intended for use on the command line, so the answer is returned as an *exit value* that can be checked to make decisions in a script. To see an

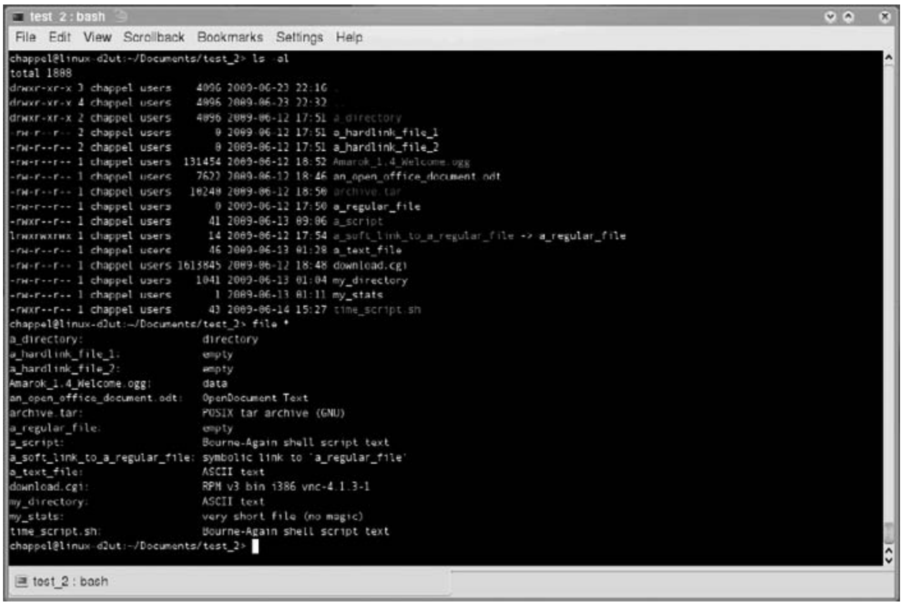
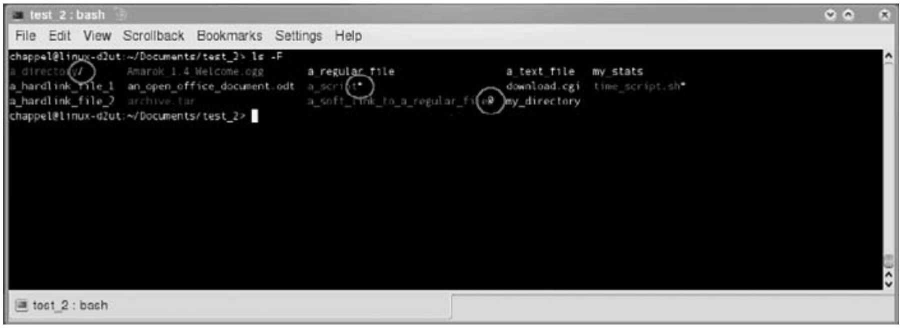


FIGURE 6.6
Example of the `file` command.

FIGURE 6.7
An example of `ls -F`*.*



exit value, try this: `test 1 -gt 2; echo $?`. A 0 means "true" and a 1 means "false." The power of the `test` command really shows with some of the more interesting options, such as `test -e [filename]` to check if a file exists to be sure not to overwrite it, and `test -d [directory_name]` to see if a directory exists. There is a *test* option to check any file attribute or to compare string values.

The `ls -F` command lists files with various file types tagged for easier recognition. Where the `file` and `test` commands are more useful within scripts, `ls -F` is more helpful for humans. Common tags are as follows:

- * executable
- / directory
- @ symbolic link
- | named pipe

The tags appear after the file names as in Figure 6.7.

Editing Files Using `vi`

Now that you've learned everything else about files, it's time to learn how to edit them. The handiest text editor you'll find is `vi`. It comes installed by default with virtually any Unix-style system, although most current Linux distributions actually have *Vim* (Vi IMproved), which is an enhanced version written by Bram Moolenaar (who hosts his project at www.vim.org). They are both accessed with the same `vi` command. The `vi` is quick and easy – once you figure it out. Unfortunately, `vi` is notoriously unfriendly for new users, but it's well worth the effort to learn.

> **Note**
> It's worth pointing out the distinction between a *text editor* and a *word processor*. Although at first glance they may look identical – they both allow the user to type

stuff that shows up on a computer screen – they are used for entirely different things. A word processor is used to type letters and things that look nice, and have lots of features for formatting and arranging information in visually appealing ways – fonts, colors, tables, and so forth. A text editor is meant for editing system files; stuff meant to be read by computers. Advanced text editing features include features that are handy for programmers, like parenthesis matching, autoindenting, and color-coded key-word (syntax) highlighted based on the programming language you are using. Like any fancy text editor, `vi` supports all that cool stuff, but it's most often used to quickly pop into a configuration file to make a small change and keep moving, so that's what we'll concentrate on here.

The key to dealing with *vi* is understanding that it has three modes: *Command*, *Ex*, and *Edit*. When first starting *vi* you'll be in *Ex* mode, which allows you to move around in the file you are editing, perform *copy* and *paste* commands and other advanced features (like recording and replaying macros). To actually enter text, you'll need to shift into *Edit* mode by typing one of the edit keys, and then go back to *Ex* mode by pushing the **Esc** key. *Command* mode is entered from *Ex* mode by typing "**:**" and is used to enter save files, enter a filename, or text to be searched for. You can go back to *Ex* mode by pressing **Esc** again.

Particularly useful movement commands in *Ex* mode are as follows:

- The arrow keys (up, down, left, and right) move in the appropriate direction, one character or line at a time. If you precede the arrow with a number, it will move that number of characters or lines. This feature will multiply the effects of most `vi` *Ex* mode functions. As a holdover from the days of serial terminals that may not have had arrow keys, you can also use the **h, j, k**, and **l** keys for left, down, up, and right.

- The **G** key (capitalized) will jump directly to the last line of the file. A number in front of the **G** will take you directly to the corresponding line number. This is particularly useful if you are trying to compile a file and the compiler gives you an error at a specific line number.

- The **D** key will delete from the current cursor location to the end of the current line, the `dd` command will delete the entire current line. A **d** followed by a movement key (that is, a down arrow) will delete the line below the current cursor location. A number in front of that will delete that number of lines.

- If you make a mistake, the **u** key will undo your last command.

- You can perform searches forward with the / key, followed by the text you'd like to search for. Use **?** to search backwards instead.
- The period (**.**) will repeat the last edit command.

To transition to *Edit* mode, use the *i* key to insert text in front of the current cursor position, or *o* to insert text onto a new line below the line the cursor is on. Once in *Edit* mode, you can type text as you normally would. Press the **Esc** key to get back to *Ex* mode.

When you are finished moving and editing, you can press the colon key (**:**) to go to *command* mode. From *command* mode you can:

- perform a search and replace by typing:

  ```
  %s/some_old_text/some_new_text/g
  ```

  s is the search function (in the forward direction), the % tells it to search all lines, and the trailing *g* makes the change globally (instead of just the first instance found).

- Save and exit the file by using `wq`. The *w* writes the file; the `q` quits. To quit without saving, use `q!`.
- Get help from within `vi` by typing **`help`** at the *:* prompt. You can add a command you would like specific help with after the `help` to go straight to that section.

You can also go through a built-in `vi` tutorial by typing `vimtutor` at a command prompt. Lots of additional useful information about more advanced features is available on the vim home page at www.vim.org.

Managing Processes

Every program, utility, or bit of code waiting in the background on your Linux machine is called a *process* and is automatically handled by the kernel. Users can monitor and manage their own processes; system administrators have access to nearly all processes. Process management tools allow you to see what your machine is doing, adjust the priority of each process, and even terminate them.

> **Note**
> You can't kill the init process. On top of that, sometimes processes get stuck and nothing you do will kill them. Finally, zombie processes are notoriously hard to get rid of. All that being said, the only process that root *can't* kill is pid 1, the init process.

The two main utilities for viewing processes are `ps` and `top`.

ps

The `ps` command all by itself will show you what programs are being run by your user in the terminal window it is executed in, which is usually just `bash` and the `ps` command itself – not terribly interesting. To really see what is going on, you need `ps -A`, which will show all the processes currently recognized by the machine. The output from the `ps` command is formatted into four columns, as shown in Figure 6.8. The first column is the process identification number or PID, which the system uses to uniquely identify each process. The next column is where the process is running – it's "control terminal," which determines where the program's input and output should go.

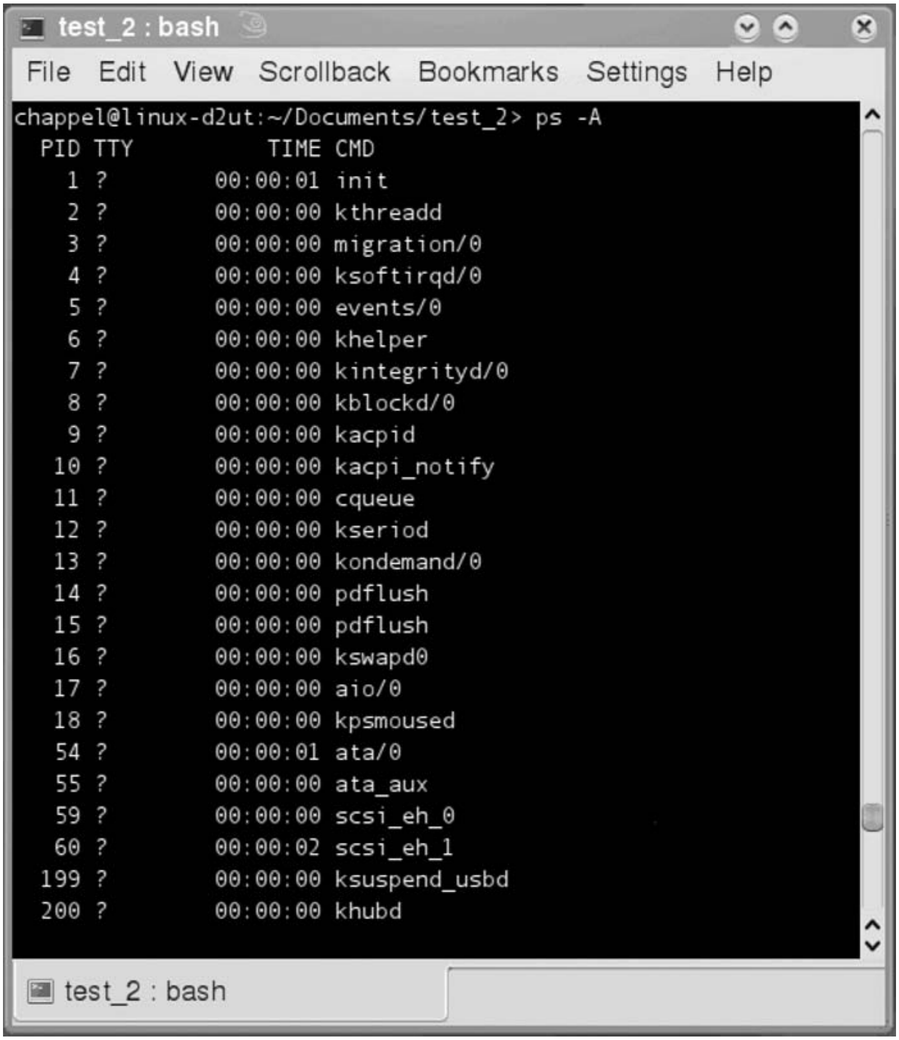


FIGURE 6.8
The `ps -A` command.

Next is the amount of time the processor has spent on the process. The last is the "human readable" name of the process.

The syntax for the `ps` command is as follows:

```
ps [options]
```

Some useful options include the following:

- `-u` or `-user` The username or user ID will show processes associated with the given user.
- `-C` The `command_name` will show processes that have the given command name.
- `p`, `-p`, or `-pid` The `process_id` will show processes with the listed process ID. If *ps* is only given a number as an argument, it assumes the number is a process ID.
- `-t`, `-tty` The `ttylist` will show processes on a given tty interface port. A plain "-" will show processes not associated with a port, and a `-t` without a tty number will assume you want the processes associated with the current port you are using.
- `-M` or `Z` (note no dash with the `Z`.) It shows extra security data associated with SELinux, discussed in Chapter 10, "Securing Linux."
- `-f` will show processes somewhat graphically associated with the parent process that started them.

kill

Occasionally a process will get stuck, or "hang," and it's necessary to forcefully end it. A command for this is `kill`. The `kill` command uses the process id, not the name – to end a process, it's necessary to use the `ps` command to find the process number, then use `kill` to terminate it. One shortcut you can use is the `killall` command, which will terminate a process by name instead of PID. However, notice that multiple instances of the same program will have unique process IDs, but with the same name, so it may be difficult to determine which one you want to end.

The `kill` command works by sending a *signal* to the process, which by default tells the process to terminate by sending the TERM signal. It can also be used to send different signals, the most common of which are subtle varieties of TERM which offer varying emphasis. The syntax for signals is `kill -s [signal] PID` – for example, `kill -s KILL 2727` or `kill -s HUP 3143`.

Signals typically used with the `kill` command, as described in "The Linux Administration Handbook,"[2] are as follows:

- `KILL` (signal number 9) cannot be blocked by the process. It essentially tells the kernel to clobber the program.
- `INT` (signal number 2) is the *interrupt* signal, the same as typing a **Ctrl + c**. It allows a user to get the attention of a program, stop what it is doing and wait for more user input, or quit gracefully.
- `TERM` (signal number 15) is a nice request for a program to finish up and quit normally.
- `HUP` (signal number 1) came from the days of serial terminals and is used to mean "hangup." Today, it is most often used to request that a program take a moment to re-read its configuration files, without actually unloading and restarting the entire program. You may occasionally see a command issued with a `nohup` option, which tells the program to keep running even if the terminal that started it is closed, so if you open a terminal window to start a service it won't quit when you close the window.
- `QUIT`(signal number 3) is the same as `TERM`, but it can cause a program to copy the memory it was using to a diagnostic file, which can be useful for troubleshooting.

> **Note**
> There is an alternate syntax that leaves off the "s," like this:
>
> `kill-HUP 5253`
>
> or
>
> `kill-INT 76`

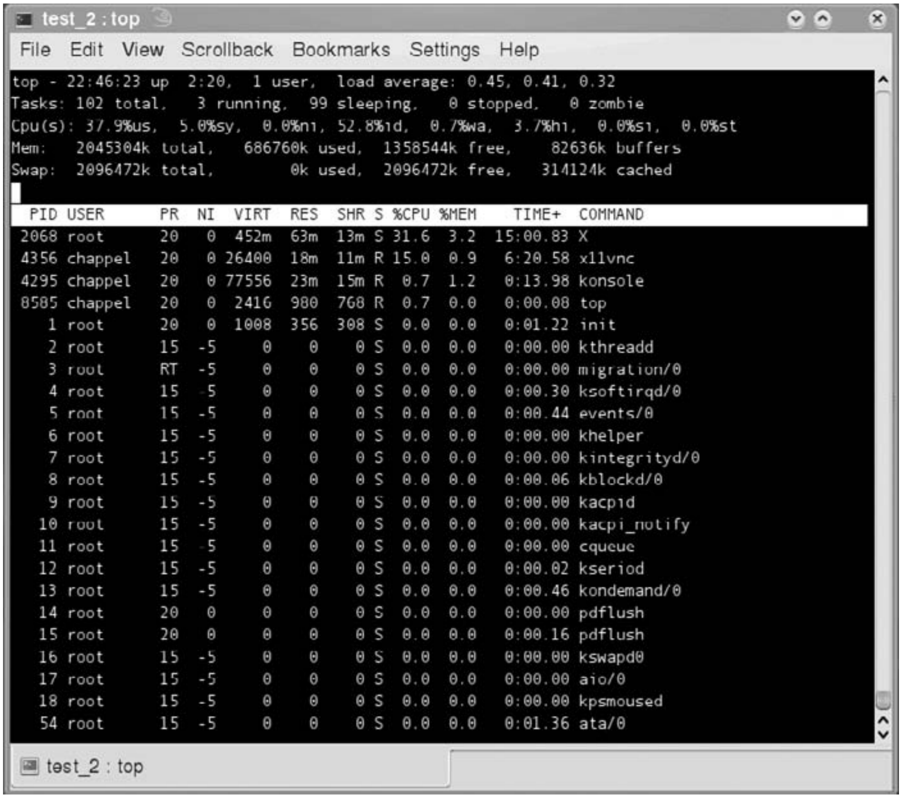
Signals can also be used to communicate between processes or by the kernel to report an event or problem back to a process.

top

A more versatile utility is `top`, which will dynamically show running processes in real time, sorted by the amount of CPU time they are using (as seen in Figure 6.9).

The upper section of the `top` display shows a summary of the system, including uptime, current users (each bash session counts as a user), and various load and memory usage statistics. The lower part of the display shows the process ID, owner, amounts of various memory, and CPU resources being

FIGURE 6.9
The `top` *command.*



consumed by each process. Additional features can be found by typing **h** for help, or consulting the `man` page (`man top`). One handy feature is a built-in `kill` option; to terminate a process, just press **k** and type in the process ID. Press **q** or **Ctrl** + **c** to exit `top`.

pstree

Another helpful way to view processes is with the `pstree` command, which shows the relationship between main *parent* processes and the *child* processes they have created in a hierarchical tree view. By default, it only shows process names, but `pstree -p` will show process IDs also. Notice that the *init* process is the parent of all the other processes – the kernel automatically starts the init process as the machine boots, and it in turn runs the startup scripts that fire up everything else. The init process always has a PID of 1. Each child process knows its parent process ID (PPID), which is what `pstree` uses to trace the relationship between processes. PPID numbers can be viewed with the `ps -Al` command.

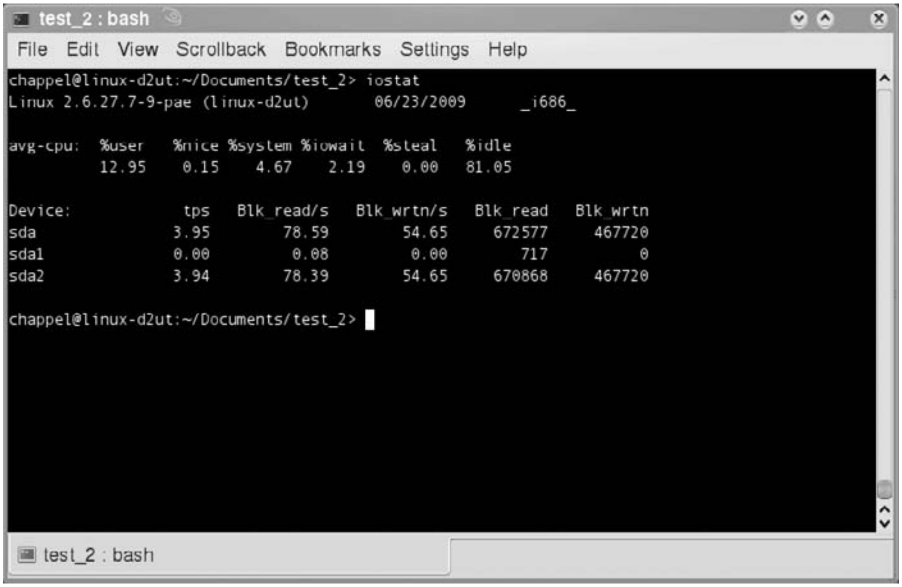


FIGURE 6.10
Example of `iostat`*.*

To look at not only CPU but also physical and network drive usage, you can use the `iostat` command, as shown in Figure 6.10.

The various CPU-related statistics shown at the top of the report are as follows:

- **%user** is the percentage of processor time spent on user applications.
- **%nice** is the percentage of processor time spent on user applications that have been allowed to run at a lowered priority.
- **%system** is the portion of processor time used by the system itself.
- **%iowait** is the percentage of time the processor has been idle when the IO system has been busy. This is a good indication that you have a bottleneck somewhere in your hard drives.
- **%steal** is related to virtualized systems.
- **%idle** is the amount of process time spent doing nothing.

The next section of the report shows statistics for drives attached to the system and includes the following:

- **tps** (transfers per second) A transfer is considered as a request for data.
- **Blk_read/s, Blk_wrtn/s** Refer to blocks read and written per second, where a block is a sector of the hard disk, usually 512 bytes.
- **Blk_read, Blk_wrtn** Refer to total blocks read and written.

Note
`iostat` isn't a part of the default openSUSE installation. To install it, type the following at the command line:

```
sudo zypper install sysstat
```

nice

Linux process controls not only let you start and stop a process, but also allow you to make it run faster or slower. On old multiuser UNIX systems, a user could set a parameter that told the kernel how and what priority to assign a process; a lower priority means the program would go slower, but the CPU would be available to run other users programs faster. The priority is a number between −20 and +19, and is called a *niceness value* (since sharing resources is nice, right?). The lower the niceness value, the less nice your process is, and the faster it runs as it selfishly hogs the system. The `ps -Al` will show niceness in the *NI* column.

The niceness of a process is inherited from the parent process. The owner of a process can make it "more nice" (making it run slower and freeing up resources), but isn't allowed to make it less nice; only a system administrator can do that. There are two ways to adjust the niceness of a file: `nice` and `renice`. To start a program with an adjusted *nice* value, type the following:

```
nice -n 10 find / -name my_file.doc
```

This will start a reduced priority file search of the entire hard drive (which will give you plenty of time to do a `ps -Al | grep find` in another terminal window to confirm if it really has a niceness of 10). Once the process is running, it can be reset to a specific value with root access and `renice` like this:

```
sudo renice 15 [PID]
```

Use the PID from the `ps -Al` command you just did. Notice that the `nice` command adjusts the niceness value by the given amount, whereas the `renice` command sets it to the exact number given. You can also adjust *all* the processes that belong to a single user instead of just a single process ID, like this:

```
sudo renice -20 -u username
```

That wouldn't be very nice at all! When you are finished playing with niceness, you might want to issue a **Ctrl + c** to end the `find` command.

Of course, you probably aren't working on an old mainframe; processors are blazing fast and most bottlenecks now are hard drives and network interfaces, so niceness doesn't get adjusted very often. Chivalry may not be dead, but niceness is definitely fading.

Leveraging I/O Redirection

One of the primary tenets of UNIX is the use of small programs that each do one thing very well.[3] These small useful tools are meant to work together to build up the exact features needed to perform whatever task you need to perform. I/O redirection is the glue that sticks these handy little programs together.

Before digging in, it's important to understand that console programs make use of three *streams* for communicating. They are shown in Table 6.1.

When using a command line program, normally you provide input through the keyboard and the program responds with output and any problems it may encounter to the console. Things get interesting when you begin to harness the power of *redirection*. This allows you to use the output from one program as the input for another, or send it to a file. You can use the contents of a file as the input for a program, or use a succession of programs to create and manipulate information. It works like this:

```
some_program > someplace_other_than_the_console
```

To see what redirection looks like, check out Figure 6.11.

Although "someplace_other_than_the_console" is frequently a regular file, since Unix treats virtually everything as a file, it's just as easy to send the information to a printer port or other "file" – even the sound card, if you'll recall our previous experiment in the "Device Files" section.

Exam Warning

Be careful when using >; if you redirect your output stream to an existing file it will be overwritten. Use ≫ to append to the end of an existing file.

Table 6.1 The Three Communications Streams

Name	Number	Symbol	Normal Connection
STDIN	0	<	Keyboard
STDOUT	1	>	Console (display)
STDERR	2	>	Console (display)

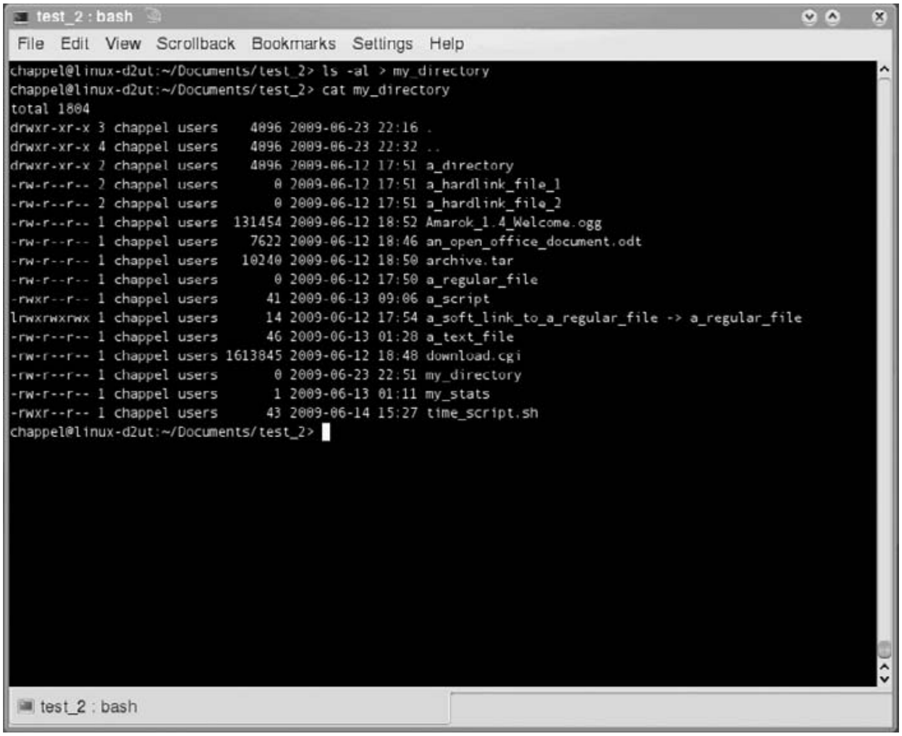


FIGURE 6.11 *Example of STDOUT redirection.*

Similarly, the stream can be reversed by using a "<" instead of ">." Note that the order of the commands doesn't change, even though the data stream now flows in the opposite direction.

```
some_program < something_to_send_to_a_program
```

An example of this can be seen in Figure 6.12.

How about STDERR? Isn't it all just output? Well, if you are redirecting the output of a program and if there is a problem, it'd be tedious to check your output file to find some error message instead of your data; so although it would normally come out on the display, it really is an entirely separate stream. Because of this, it's possible to have all the data go to a file and the errors still come to the screen, or to redirect the errors to either the same or a different file using the stream number like this:

```
some_program > data_output.txt 2> error_output.txt
```

To redirect only the error stream, do this:

```
some_program 2> error_output.txt
```

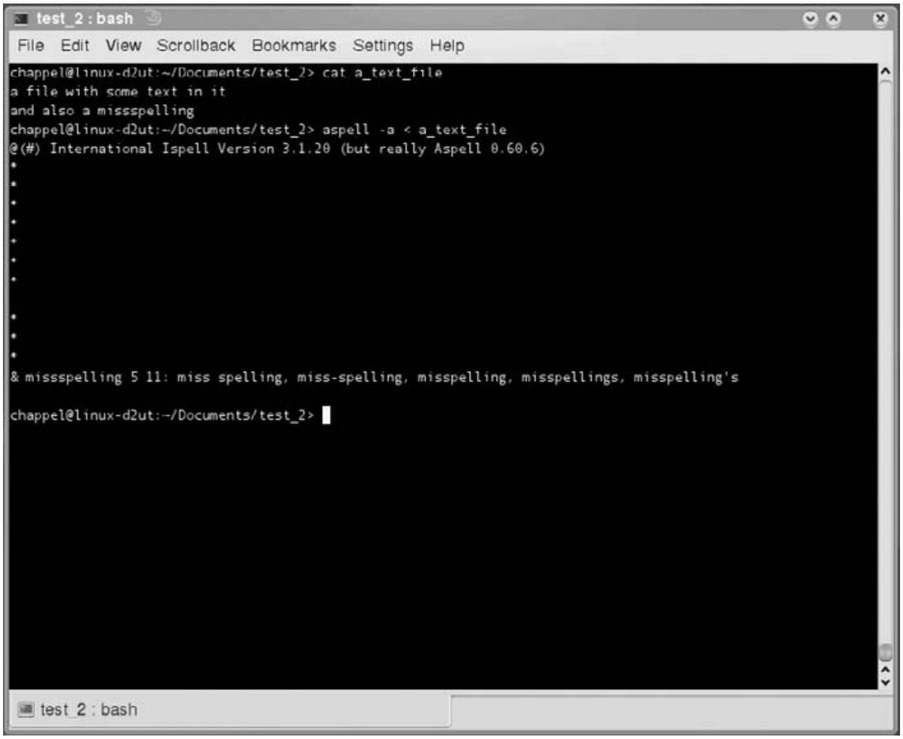


FIGURE 6.12 *Example of STDIN redirection.*[4]

To combine the data and error, use either this:

```
some_program > data_and_errors.txt 2>&1
```

or this shortcut:

```
some_program &> data_and_errors.txt
```

To send the output from one program into the input of another, you *could* use a convoluted combination of > and <; but UNIX provides the `pipe` command, |, which is much easier. The pipe works like this:

```
some_command | another_command
```

For example, you can list the running processes, and then search the list for all instances of Firefox:

```
ps -A | grep firefox
```

If you are of the sort who likes to have your cake and eat it too, the `tee` command will duplicate a stream and send one copy to the display and another to a file. It's used with the `pipe` command:

```
ls -al | tee directory.txt
```

A potential problem with piping the output of one command into the input of another is that some commands will only accept a certain amount of input at a time, and may complain if they have massive outputs thrown at them. The solution to this is the `xargs` command, which will accept the input stream and split it into chunks to pass along a bit at a time and repeatedly running the downstream command until the input stream ends. It is often used with `find` to run a command for each item found. The following example will move all of Bob's music files into his music directory:

```
find . -name "*.mp3" -u bob | xargs -i mv {} ./bobs_mp3s
```

There are other ways to string commands together, too. By putting a ";" between two commands, they will run consecutively – when the first one is finished, the second one will start:

```
do_this_first; then_do_this; and_finally_this
```

Note that spaces are ignored, and can be used before, after, both, or not at all.

> **Note**
> Two interesting variations of the *;* function are *&&*, which only runs the next command if the first one completes without an error, and || runs the next command only if the first one fails – handy for sending a message to notify someone of the problem.

The "=" is used to set a variable equal to a value using the format `name=value`, as shown in Figure 6.13.

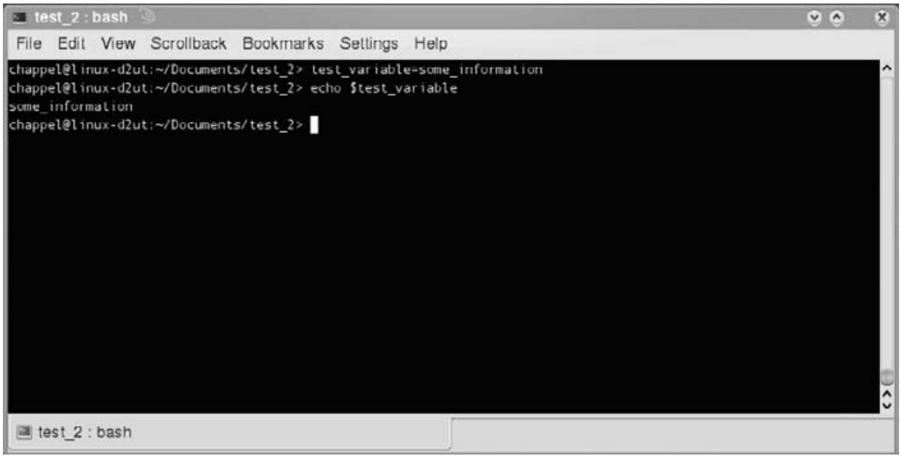


FIGURE 6.13
An example of setting a variable equal to a value.

Variables are typically used within scripts, as in this simple backup script from the Linux Documentation Project:[5]

```
#!/bin/bash
     SRCD="/home/"
     TGTD="/var/backups/"
     OF=home-$(date +%Y%m%d).tgz
     tar -cZf $TGTD$OF $SRCD
```

The "==" is used to compare the equivalence of two variables. An example is comparing two values in *bc* – a command line calculator that, among other things, will return a 1 for a true comparison or a 0 for a false one:

```
bc
10==10

1

10==1

0
```

Learn by Example: Optimizing Frequent Tasks

Like any operating system, Linux offers many ways to automate common tasks. I frequently ping my default gateway of 10.10.10.1 but get tired of typing it out all the time, so I've added the router's name and address to the end of my /etc/hosts file, and now Linux will resolve it by name.

One way to do this is to use *vi* to edit the file. A *G* takes me straight to the end of the file, and an *o* adds a line to the bottom; type the address and name, and a quick `<Esc>:wq` and I'm done. Even quicker, though, is to use `echo "10.10.10.1 router" » /etc/hosts`, which automatically appends the information I need at the end of the file. Just have to make sure I use two greater-than signs – just one will overwrite the file completely.

I also use my laptop on a lot of different networks, and don't know the default gateway ahead of time. I can build that right into the command, though, using an alias and some command-line redirection:

```
alias pdg='ping -c 3 ‘ip route show 0.0.0.0/0 | cut -d " " -f 3‘'
```

The `ip route show` gives information about the default route; but I only want the actual IP address, so I use `cut` to grab just the third piece of information in the list (the address). All that is inside the tick marks (‘), so it gets worked out and then fed to `ping`. I could just type that whole thing out every time, but that's even worse than

doing it by hand – so I use the `alias` command to create a whole new command, `pdg` for "ping default gateway."

```
chappel@agatha:~$ pdg

PING 192.168.10.1 (192.168.10.1) 56(84) bytes of data.

64 bytes from 192.168.10.1: icmp_seq=1 ttl=64 time=0.686 ms

64 bytes from 192.168.10.1: icmp_seq=2 ttl=64 time=0.484 ms

64 bytes from 192.168.10.1: icmp_seq=3 ttl=64 time=0.484 ms


--- 192.168.10.1 ping statistics ---

3 packets transmitted, 3 received, 0% packet loss, time 1998ms

rtt min/avg/max/mdev = 0.484/0.551/0.686/0.097 ms

chappel@agatha:~$
```

Special Devices

You may have noticed some particularly unusual files when peeking into the /dev subdirectory:

- /dev/null
- /dev/random
- /dev/zero
- /dev/urandom

The /dev/null file is often called *the bit bucket*, and is like having your own personal black hole. Any output you don't need to see? Ever? Just redirect it to the bit bucket. This is most often used to hide unnecessary output from a program embedded in a script.

The /dev/random file does just what it says: it's the interface to the systems random number generator and will produce random numbers used to create very secure cryptographic keys. Creating true random numbers is difficult for a computer, though. Current Linux systems use noise from various device drivers and other nonrepeating information to maintain an "entropy

pool" used to generate the numbers; but it's possible for the pool to run dry – so to speak. If this happens, /dev/random blocks further requests until more noise can be collected. Applications that can get by with *psuedo-random* numbers can use /dev/urandom, which could theoretically be vulnerable to attack; but is still pretty good, and won't run out.

The /dev/zero file works like /dev/null if you send data to it, but you can use it for an input stream to create files full of zeros.

> **Note**
> Another interesting special device file is /dev/full, which acts like a full device. This is handy for testing programs to see how they fail when the drive to which they are trying to write is full.

Using System Documentation

Easily the most important thing you can learn is how to learn more. Linux offers many ways to find additional information for all of its commands. The quickest and most basic assistance comes from using the help option of any given command – usually both `-?` and `--help` work, as in `ls -?` or ls `--help`.

man #

If you need a little more detail, check the `man` pages (short for *manual*). `Man` pages are brief descriptions of a command and all its available options, and are accessed by typing `man [command]`, which then goes into a full-screen viewer. Try `man man` to get the full tour. Briefly, <**Spacebar**> or <**Page_Down**> goes through the manual entry one page at a time, <**Page_Up**> goes up a page, / lets you enter text to search for within that entry, and `q` (for quit) exits back to the command prompt. If that sounds familiar, it's because `man` uses `less` as a `pager`, so it's the same interface you'd get if you typed `ps -A | less`.

The actual information shown by `man` lives in a couple of different directories. These vary by distribution, but the main information is in /usr/share/man or /usr/share/doc. The `manpath` command will show them to you exactly where `man` looks for information on your system. When you install a new program using a package manager, it will usually take care of adding the appropriate manual pages – usually to /usr/local/man.

If you look in one of the manual directories you'll see the pages are split up into numbered groups, called *sections*. The most useful information for system administrators is in sections 1 and 8. The section numbers are shown in Table 6.2.

Table 6.2 The Various `man` Page Sections

Section	Information
1	User commands
2	Kernel functions
3	Library functions
4	Devices, special files in /dev
5	File formats
6	Games
7	Miscellaneous
8	Root level system admin tools
9	Kernel specs and interfaces

Cross-references to other man page often give the section in parenthesis following the command, such as `ls(1)`.

apropos

What if you aren't sure what command you want? You can use `apropos` to search through the descriptions of each man page to help find what you want. For example, if you are searching for a file on your computer and can't remember what command to use, try `apropos search` to see all the manual page entries that have *search* in their description. You can also send `apropos` wildcards, use regular expressions, and limit it to specific sections; go to `man apropos` for details. Note that `apropos <keyword>` and `man -k <keyword>` do the same thing.

whatis

If you have a command but aren't sure what it does, you can use the *whatis* command to print out just the one-line description from the commands man page. Try `whatis ls`.

Behind the scenes, `apropos` and `whatis` don't actually check every man page entry; they scan through a database that gets created by the `mandb` program, which is run regularly in the background by a `cron` job to keep things current.

makewhatis

The database created by `mandb` may be handled by `makewhatis` on some distributions.

`makewhatis`, like `mandb`, runs occasionally and creates a file containing all the short command descriptions for `apropos` and `whatis` to search through.[6] Keep in mind that if you choose to try running `makewhatis` manually then it needs to be run with administrator privileges. You can use the `-v` option to have it print out a more detailed description of what commands it is indexing.

info

Another way to get information about programs is with `info`. Whereas `man` is a universal holdover from UNIX, `info` is a creation of the GNU project, and is intended as an improved way to manage documentation for all the GNU utilities.

The `info` program provides information in threaded chunks called *nodes*. Each node contains a description of a particular program at a specific level of detail, so a complex program may have a number of nodes. Typing **`info`** all by itself takes you to the top level node; typing **`info program`** takes you to the information node for that program. If there isn't an `info` page available for the program, `info` will search for a `man` page, and if that fails, it will take you to the top level index node.

Because of this extra level of features, there is an extra level of commands. There is a helpful tutorial available by typing **`info info`** at the command line. An `info` screen looks like Figure 6.14. Notice how the first line shows you which node you are currently viewing, and which nodes (may be) next, previous, and up from your current node. The bolded line at the bottom also shows which node you are viewing and where you are within it (Top, a percentage through it, or Bot if you are at the bottom).

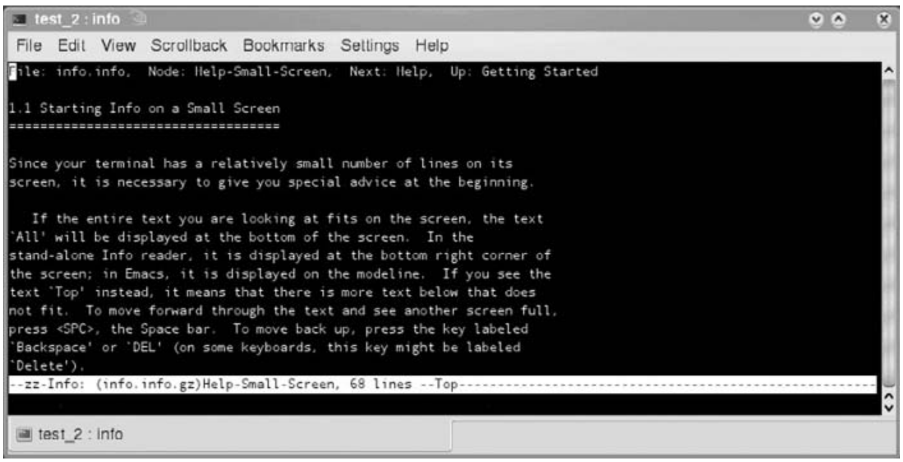


FIGURE 6.14
An example of an `info` screen.

There are a bewildering number of available commands within `info`, but the basics are as follows:

- The arrow keys move up and down by a single line within a node (page).
- The **Spacebar** and **Page Down** keys move down the node a screen at a time, and will jump to the next node when you reach the bottom.
- The **Backspace** and **Page Up** keys will move up the node a screen at a time, and will jump to the previous node if you are at the top.
- If there are a series of nodes, *n* will move to the next node and *p* will move to the previous one.
- The **Tab** key will jump between hyperlinks within a node, and the **Enter** key will follow the link.
- When finished, use the ***q*** key to quit `info`.

Exam Warning

Remember that when you are taking the Linux+ exam, you'll have to *know* about the Linux documentation systems – but unfortunately, you won't have access to them.

Using Virtual Consoles

Linux, having its roots in giant multiuser UNIX systems, is designed to handle many users at the same time. That isn't generally how PCs are used now, but it's still handy to have the option. Each Linux system runs seven simultaneous virtual consoles – just like there were eight screens connected to the computer all at the same time. From the standard Xwindows graphical environment press and hold **<Ctrl><Alt>** and then **<F1>** through **<F7>** to access the virtual consoles screens 1 – 7. Note that screen one has the boot log information – if you want to monitor what the system is doing as it boots and it only shows you a pretty boot screen, you can press **<Ctrl><Alt><F1>** to see what is really happening. The boot log information is also available in the /var/log/ directory – see Chapter 11, "Troubleshooting and Maintaining Linux" to read about analyzing system logs. The Xwindows system runs in screen seven, so pressing **<Ctrl><Alt><F7>** brings you back to the GUI environment, if one is running.

The single handiest use for virtual consoles is to kill a hung application that has somehow messed up your GUI. Simply pop over to a free virtual console, log in, run `top` or `ps -A` to identify the offending process, `kill` it and `pop` back. If you aren't certain which process to kill, it pays to be prepared to reboot.

Accessing Kernel and Architecture Information

What is commonly referred to as "Linux" is actually an amalgam of different components. Only the central core – the *kernel* – is "true" Linux. Much of the rest, including the all-important C compiler **gcc** and most of the user programs, have been provided through the GNU project and the Free Software Foundation. Because of this, many people prefer the more technically accurate (but hopelessly clunky) name "GNU/Linux."

So, what is this Linux kernel? The *kernel* is a program that is started when a computer boots. It handles the coordination of all the system resources – allocation of memory, determining when to allow which program to be run by the processor, and managing communications between all the peripheral devices that are attached with the help of device drivers.[7]

Linux uses a *virtual file system* located at /proc to represent activity within the computer, and allow communication with various kernel and driver components. If you compare `ps -A` with `ls /proc`, you'll notice that every process has its own subdirectory under /proc. There are also files for system hardware: `cat /proc/cpuinfo` will tell you anything you could ever want to know about your processor, and /proc/version knows all about your installed version of Linux. A slightly different way to check on your installed version is `uname -a`. Note that /proc/version will show the actual distribution name, whereas `uname -a` will show if you have a 32- or 64-bit processor.

> **Note**
> The `cat` command is short for "concatenate," which is intended to join two files into one. However, if you only give it one file name, it will "concatenate" it onto the screen so you can read the contents.

One of the benefits of the Linux kernel and /proc filesystem is the ability to make adjustments while the system is running. All the settings in the proc/sys subdirectory tree can be adjusted. While you can adjust them just as though there were files, the `sysctl` command provides an easier interface. Changes made within the /proc/sys virtual file structure are lost when the system reboots, though. To make them permanent, add them to the /etc/sysctl.conf file, which gets loaded each time Linux starts up. Use `sysctl -p` to get the system to re-read the sysctl.conf file without having to reboot.

A common example of using this mechanism is to enable packet forwarding. This turns a Linux PC with two (or more) interfaces into a router, allowing it to pass traffic between two networks. The virtual file is

/proc/sys/ipv4/ip_forward. If the file contains a 0, traffic doesn't get passed; if it's a 1, it does. So to enable packet forwarding you could:

```
echo 1 > /proc/sys/ipv4/ip_forward
```

This will put a "1" in that file. Or, type:

```
sysctl -w ipv4.ip_forward=1
```

or

```
vi /etc/sysctl.conf
```

Either add or uncomment (delete the leading #) a line that says `ip_forward=1`.

That last option will make the change permanent.

Basic Scripting

A *script* is a plain text file containing a collection of command-line programs and options that can be all run as a group. Each line of the script is read one at a time and *interpreted*, which can make scripts slow compared with the other options for creating programs – but since they are made from the same commands you use daily on the command line, they are quick and easy to create. A comprehensive guide to scripting would fill an entire book; for the sake of the Linux+ test, there are only a couple key points to know.

Any file that you want to run in Linux – scripts included – need to have the *execute* permission bit set. This is covered in more depth in Chapter 10, "Securing Linux," under the "`chmod`" section. Briefly, `chmod u+x` filename will set the execute permission bit for the owner on a given file, and allow it to be executed.

The exception to this is if you use the `sh` command to kick off a new shell, which makes the `sh` program the executable and the script just a parameter, as shown in Figure 6.15.

Notice in Figure 6.15 that we start with a regular file – there are no "`x`"s in the permissions shown with an `ls -al` command. As a result, attempting to execute the script directly by typing **`./a_script`** fails, but typing **`sh ./a_script`** works fine. Once `chmod u+x` is used on the script, `ls` will shade it green for you, so you can quickly tell that file is executable.

You can also use the `bash` command in place of `sh`.

> **Note**
> If you set your script to executable and try and run it, be aware that (unlike DOS and Windows) Linux does *not* include the current directory in the path searched to

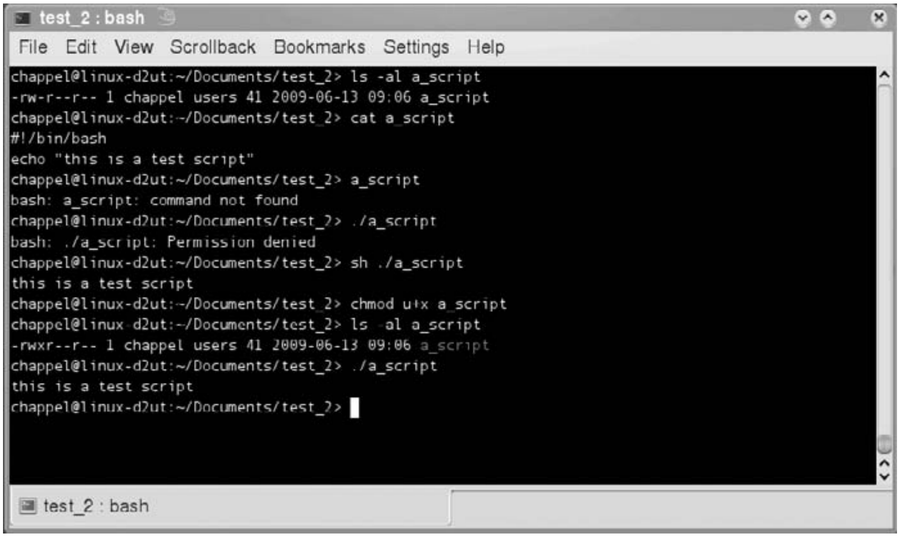


FIGURE 6.15 *Executing a script using* `sh`*.*

find executable files. As a result, you'll have to provide an explicit path to your script, although you can use the ./ shortcut – like this: `./myscript`

A bash script file expects to have a first line of: `#!/bin/bash`. Normally, a line starting with a "#" is ignored and used for comments; but this special instance tells the command interpreter that this is the start of a shell script, and this script would like to be run with the bash shell located in the /bin directory. This will ensure that your script will be run in the correct shell, even if the user that is using it is in a different shell.

The command interpreter itself is the command line you've been using, but it's also a programming language. Once you've mastered using it interactively, there is a whole additional world of creating scripts that can start doing your work for you.

Using Shell Features

The bash shell provides a history feature that remembers what commands you've previously entered. The list of commands is stored in a hidden file in your home directory: .bash_history (note that hidden files start with a "`.`" and can be listed by using `ls -a`). Because the history is stored in a file, it doesn't go away when you log out or reboot, although it does change when you log in as root since you aren't the same user anymore. You can view the entire list by typing `history`. Recall that output of one command can be

piped to another; a convenient use for this is that it lets you search for a past command using `grep`. To find all previous `ps` commands, you've used for example, you could enter `history | grep ps`. Well, not *all* – by default, it backs only 500 commands.

The `bash` history feature lets you hit the **Up-arrow** key to cycle through your most recent commands, but that's just scratching the surface of what it can do. When you type `history`, you'll notice each command line is preceded by a number; typing `!<command_number>` you can replay the command that corresponds to that number. Typing `!!` will repeat the last command, if you don't like the **Up-arrow** for some reason. If you made a mistake or want to retype a command and only change a small bit, you can do a quick substitution of text in the last command like this: `^old_text^new_text`. For a complete description, check out `man history`.

Another handy command-line feature is tab completion. `bash` will try and guess what you are attempting to type if you push the <**Tab**> key. If you already have enough characters to uniquely identify the next word, bash will just fill it in for you. This works for programs, files, and directories. If you haven't typed enough of a word, a second push of the <**Tab**> key will display a list of possible matches for you. It will even warn you if the list is very long. Apart from saving a lot of typing if you have to deal with long file names or directories, it's great for avoiding typos, too.

SCHEDULING TASKS

Running scripts to automate complex tasks is great, but it really turns into magic when you can make the scripts run themselves. No one wants to log in at 3:00 A.M. to start a backup job, especially if it has to run every evening. In the sections below, you will learn how to put routine tasks on an automated schedule using `cron` and how to schedule ad hoc tasks at the most convenient time using `atq`.

cron (cron allow, cron deny)

For the routine tasks that need to be performed on a regular basis or at predefined intervals, Linux provides a scheduling service called `cron` that lets you put these tasks on a schedule of your own design. Daily backups to disk or tape is a prime example of a routine task that does not require your active supervision. The `cron` program starts up when the system boots and runs in the background, checking every minute to see if there is anything scheduled to run.

For security purposes, there are filters available to limit who can edit the *cron* configuration (*crontab*) files. Users in the /etc/cron.allow file are the *only* users able to edit their files. If that file is empty, the system looks for users in the /etc/cron.deny file, which lists users barred from editing *cron* files. The root user isn't affected by either.

`crontab` Command Syntax

The schedule consists of a series of files called *crontabs* (cron table) that tells `cron` what to run and when to run it. Crontab files for users are stored in /var/spool/cron, with the crontab name matching the username. The administrators files are kept in /etc/crontab, and there is an /etc/cron.d directory that programs can use to store their own schedule files.

The crontab schedule files are edited using `crontab -e`. They use a special format to specify the when to run. Each line has either six or seven fields, and specifies one command. The first five fields on each line tell which minute, hour, day, month, and weekday to run a command. The next spot on a line is an option for administrators and tells whose user rights to run the command under, and last is the command itself. Valid options for the schedule fields are a `*`, which matches anything, a number, which matches that exact value, two numbers with a dash between them for a range of values, or a list of values separated by commas to match each single value. The command can be anything that works. Altogether, it looks like this:

```
crontab -e

30 17 * * 1-5 do_something.sh
```

That means 30 min after the hour of 5:00 P.M. every day of the month, every month of the year (that is, Monday through Friday).

`crontab` uses the text editor that your profile is set to use; chances are you'll be using your old friend `vi`. Remember to press ***i*** to insert text, **<Esc>** to get back to ex mode, then **:**`wq` to save and exit. See `vi` section if you need a refresher.

You can also use `crontab -l` to display the contents of your crontab file, and `crontab -r` to delete it.

atq

If you just want to have something run a bit later rather than schedule it to keep reoccurring, you can use the `at` command. It accepts commands from either a file (use `-f`) or standard input, ending with a `<ctrl> d`. A single `ctrl-d` is sufficient if there is nothing else on the line. It takes two `ctrl-ds` if there is more text on the line. For example, `at> ls -l >text.txt`

`ctrl-d ctrl-d`. Use the `atq` command to see what jobs are queued up, and their job numbers. To cancel a job, use the `atrm` command, followed by the job number you want to cancel.

EXERCISE 6.2: Scheduling a Task for Later

In this exercise, we will demonstrate using the `at` and `wget` commands to download a file after-hours, so it doesn't interfere with daytime Internet usage. To do this, use the following commands:

1. Find the file you'd like to get. In Firefox, this is done by right-clicking and selecting **copy location** instead of clicking on the file to download it.
2. At the command line type `at 01:00` to schedule your download for 1:00 A.M. the next morning. When you press **Enter**, `at` comes back with a > prompt to ask what you'd like it to do at 01:00.
3. At the prompt, type `wget <space>` and `<Ctrl><Shift>v` to paste the file location you copied from Firefox.
4. Type `<Ctrl>d` to end. (You may need to enter `<Ctrl>d` twice.) If something goes wrong, you can type `<Ctrl>c` to cancel.
5. Type `atq` to confirm you have a job queued up.

Now your selected file will download as the user you were logged in as, into the directory you were in when you invoked `at` (`at 01:00`), when hopefully no one will complain that you are hogging all the bandwidth. ■

MANAGING SERVICES

As a system administrator, one of your primary jobs is to manage services running on your systems. This involves installing and starting new services, reconfiguring and restarting services to make changes, and stopping services that may be having trouble. This section will describe how to perform these tasks through start-up scripts and from the command line.

/etc/init.d

Services, or daemons, are just programs that the system starts itself. They run in the background and provide services for users, such as the Apache Web server or Postfix e-mail server. As described in the "Runlevels" section of Chapter 4, "Booting Linux," services are started when their assigned runlevel

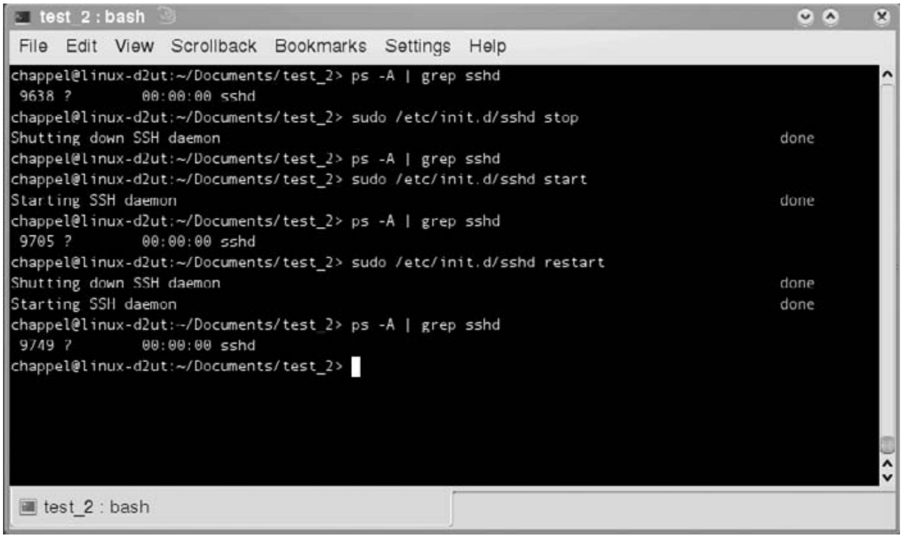


FIGURE 6.16
An example of stopping, starting, and restarting a service using init *scripts.*

is initiated. The scripts that initially start each service can also be used to manage them manually, and can be found in the etc/init.d directory. The scripts accept a parameter of either `start`, `stop`, or `restart`. Some start scripts support an additional command, `status`, which displays the status of the service (running or not running). As an example, to restart the secure shell service (sshd) after making a change to its configuration file, you'd type */etc/init.d/sshd restart*, as seen in Figure 6.16.

`inetd` and `xinetd`

To conserve resources, some smaller and less frequently used network-based services get bundled together in a *super server* service, which waits in the background until one of the services is needed and then loads it. This method prevents the service from just sitting there taking up memory until it is actually needed. There are two super server daemons in Linux – `inetd` and the newer, "extended" `xinetd`, which adds some features and enhanced security. Changing the services managed by `inetd` or `xinetd` is conveniently done through GUI tools on current Linux distributions (YaST|Network Services in SUSE); it can also be done manually by editing the appropriate configuration file (/etc/inetd.conf or /etc/xinetd.conf) and restarting the service (`kill -s SIGHUP`).

chkconfig

A handy tool for managing which runlevels a service runs in is `chkconfig`. It can be used to view or change the runlevels that a service will run in. Type `chkconfig` to view the status of all services at the current runlevel.

Some of the options for `chkconfig` include the following:

- `-A` or `--allservices` displays all services; same as `chkconfig`, with no options.
- `-t` or `--terse` [name of service] will show if the named service is set to run at the current run level. Note that it doesn't actually show if the service is running, only if the init scripts indicate that it *should* be running.
- `-l` or `--list` [name of service] will show the init script settings for the named service for all run levels.
- `-a` or `--add` [name of service] will set the scripts to run the named service at the given run level.
- `-s` or `--set` [name of service runlevel(s)] will set the script to run at the listed run levels. For example:

`chkconfig -s sshd 35` sets the secure shell service to run at runlevels 3 and 5. `-d` or `--del` [name of service] will set the named service to *not* run at the current run level. Note that this won't actually *stop* the service if it is running, only change the init scripts to not start it the next time they run. Please refer to `man chkconfig` for additional information.

EXERCISE 6.3: Enabling NTP with *chkconfig*

In this exercise, we will turn on the Network Time Protocol service, which keeps the system clock synchronized with a master clock over the Internet. To do this we need to do the following:

1. Well, in openSUSE we *could* just click on **YaST** | **Network Services** | **NTP Configuration**, select **now and on boot** and add a nearby time server from the list, but what fun would that be? If you are going to be an expert, you'd better not be dependent on a specific distributions GUI tools.

2. First you need to find a local NTP server. Technically, it doesn't have to be local to work, but minimizing how much network you chew up makes you a good netizen. It is preferable to have a local clock master. Otherwise try and find something in your neighborhood at www.pool.ntp.org.

3. You'll need administrative rights to do this. Either type `su` to open a root level BASH shell, or get used to typing `sudo` in front of each command.

4. Once you've picked an NTP server, add it to your workstation using the `rcntp` command:

```
rcntp addserver <server_name>
```

5. Now you can start the service by typing the following:

```
sudo /etc/init.d/ntp start
```

6. Typing `ps -A | grep ntp` should confirm the service (`ntpd`) is running; typing `cat /var/log/ntp` will show you how well it's doing. There should be some synchronizing going on.

7. To keep from having to start `ntp` by hand all the time, now, you'll want to tell the system to run it automatically:

```
chkconfig -a ntp
```

8. Now your Linux machine is automatically syncing time. Nothing to it! ■

SUMMARY OF EXAM OBJECTIVES

In the BASH and command line tools section, we covered enough to get you started creating, editing, moving, and deleting files and directories, as well as using some of the tools that are available to find files within a directory structure. We also covered using `man` and `info` to find more information about command-line utilities. We even learned about linking commands together to get even more functionality out of them, and touched on how scripts work.

In the "Scheduling Task" section, we learned how to get programs to run automatically on a regular schedule with `cron`, or just at a later time with `at`. In the "Managing Services" section, we learned how to tell Linux which services to run depending on the current runlevel, and how to start, restart, and stop services using the scripts in /etc/init.d. Scheduling maintenance tasks and getting services to run at the proper times and in their proper contexts is critical for maintaining a healthy and secure Linux system. Task scheduling will reduce the time you take caring and feeding your systems, and shutting down unnecessary services is one of the best things you can do to prevent your systems from being compromised.

To be successful on the exam, you should become comfortable using all of these commands, so that you can recall their respective purposes and syntaxes when asked. I recommend choosing the command line to perform system tasks over their GUI-based counterparts, whenever possible.

SELF TEST

1. You need to check the configuration file for your Network Time Protocol service. You know all the configuration files are somewhere in the /etc folder, and that the file would be called "ntp-something" – maybe ntp, ntpd, ntp.config, ntpd.conf.... but you aren't sure. What's the best way to find your configuration file?
   - **A.** whereis ntp*
   - **B.** ls /etc/ntp*
   - **C.** find /etc -file ntp*
   - **D.** ls /etc/ntp?
2. You want to show the files in your current directory sorted by date, but you aren't sure which option for *ls* is correct. How could you find out?
   - **A.** `help ls`
   - **B.** `info ls`
   - **C.** `man ls`
   - **D.** `about ls`
3. You find a file in your documents directory named *myfile*, but you don't remember what it is. Which is a good way to learn more about it?
   - **A.** `test myfile`
   - **B.** `check myfile`
   - **C.** `file myfile`
   - **D.** `info myfile`
4. Your Linux machine seems to be running slowly, and you suspect there is a program that is keeping it busy. What is the best way to check for a program that is using a lot of system resources?
   - **A.** `top`
   - **B.** `iostat`
   - **C.** `ps -A`
   - **D.** `view`

5. You have a script called my_script you'd like to run every Sunday night at 8:30. You type in crontab -e to edit your crontab file. What would the correct entry in your crontab file look like?
   - A. 8 30 * * Sun my_script
   - B. 30 8 * * Sun my_script
   - C. 30 20 * * 1 my_script
   - D. 20 30 * * Sun my_script

6. You just got a new gps-based Network Time Protocol server, so you no longer have to mooch off some university over the Internet. You edit the appropriate config file to add the IP address of your server. What do you need to do next?
   - A. Nothing. The ntp service automatically detects the change and will start using your new time source.
   - B. Use chkconfig -d ntp to turn off the service, then chkconfig -a ntp to turn it back on with the new settings.
   - C. Use the init scripts to stop -/etc/init.d/ntp stop – then start - /etc/init.d/ntp start – the service with the new configuration.
   - D. Use the init scripts to restart the service -/etc/init.d/ntp restart.

7. You are logged into a Linux computer in a windowed (GUI) environment as user jim and open a terminal (console) session. What directory will you start out in?
   - A. /home/jim
   - B. /user/jim
   - C. /jim
   - D. /pwd/jim

8. You just finished up a project and have all your files in a folder called *project1*. You archive the folder to a CD and now want to delete it from your computer. Which command will get rid of it for you?
   - A. rmdir project1
   - B. rm project1/*
   - C. del project1
   - D. rm -r project1

9. You need to make a quick update to your /etc/hosts file, so you open it in vi and make your changes. Now you are ready to save the file and exit. What do you do?
   - A. type <Ctrl>c
   - B. type <Ctrl>d
   - C. type <Ctrl>z
   - D. type <Esc>:wq

10. You want to learn more about all the hidden files in your home directory. What command could you use to see them?

 A. ls -A
 B. ls .*
 C. ls -a
 D. ls .

11. You are documenting your system and want a file named *user_dirs* that contains a current list of all the user home directories. What is a quick way of doing this?

 A. echo /home > user_dirs
 B. ls /home > user_dirs
 C. ls /home » user_dirs
 D. cp /home » user_dirs

12. You are getting hungry and you can't believe it isn't lunch time yet. You want to check that the NTP process is really running and that your computer clock isn't slow. What command will confirm that ntp is running?

 A. ps -A | grep ntp
 B. ps -C ntp
 C. ps ntp
 D. /etc/init.d/ntp status

13. You have a script name some_program that needs to start in 30 min, and you decide to try the *at* command instead of setting the alarm on your watch to remind yourself. What is the correct syntax?

 A. at 30 <return> some_program <return> <ctrl>d
 B. at now + 30 minutes <return> some_program <return> <ctrl>d
 C. at 30 minutes some_program <ctrl>d
 D. at now + .5 hours <return> some_program <return> <ctrl>d

14. You are copying a bunch of files from ./temp to ./new_stuff, but you accidentally typed cp temp/* new-stuff instead of new_stuff. You've been reading up on the command history function and want to use that to re-enter the command correctly. What do you type?

 A. ! -change new-stuff new_stuff
 B. !!s/new-stuff>new_stuff

C. `^-^_`
D. `history -r new-stuff new_stuff`

15. You just made a new script `my_new_script` and you want to run it. You remember to give an explicit path to it when you execute it by typing `./my_new_script`, but you only get an error saying you don't have permission to run the file. You remember that you need to give yourself execution rights to the file. How do you do that?
    A. `chmod 744 my_new_script`
    B. `chmod 666 my_new_script`
    C. `chmod u+w my_new_script`
    D. `chmod g+x my_new_script`

SELF TEST QUICK ANSWER KEY

1. **B**
2. **B** and **C**
3. **C**
4. **A**
5. **C**
6. **C** and **D**
7. **A**
8. **D**
9. **D**
10. **A**, **B**, and **C**
11. **B**
12. **A** and **D**
13. **B**
14. **C**
15. **A**

ENDNOTES

[1] Lucas G, director. *Star Wars Episode IV: A New Hope* [DVD]. 20th Century Fox: Lucasfilms LTD; 1997.

[2] Nemeth E, Snyder G, Hein T. The Linux administration handbook. 2nd ed. Upper Saddle River, NJ: Pearson; 2007. p. 59.

[3] Raymond E. The art of UNIX programming. Verlag: Addison-Wesley; 2004. p. 1–27.

[4] ldp. The Linux Documentation Project; simple redirection, <http://tldp.org/LDP/intro-linux/html/sect_05_01.html section 5.1.2.3>; 2008 [accessed 06.11.09].

[5] ldp. The Linux Documentation Project; BASH Programming, <http://tldp.org/HOWTO/Bash-Prog-Intro-HOWTO-12.html>; 2000 [accessed 06.12.09].

[6] Bandel D. apropos, whatis and makewhatis. Linux Journal, <http://www.linuxjournal.com/article/1329> 9/1/1996; [accessed 06.12.09].

[7] Bovet D, Cesati M. Understanding the Linux Kernel. 3rd ed. Beijing; Cambridge, MA: O'Reilly and Associates; 2001. p. 1–34.

CHAPTER 7

Installing Applications

Exam objectives in this chapter

- Install, Remove, and Update Programs
- Resolving Application Dependencies
- Adding and Removing Repositories

UNIQUE TERMS AND DEFINITIONS

- **Software package** Software packaged in an archive format that is installed, managed, and removed using a package management system or stand-alone installation software.
- **RedHat package manager (RPM)** A software package management tool, developed by Red Hat, that is used on a variety of Linux distributions; the RPM file format is the chosen standard package format for the Linux Standard Base.
- **Advanced packaging tool (APT)** A front end to the core package management system on Debian-based Linux distributions (dpkg) to install, manage, and remove software packages.

INTRODUCTION

By the time you've landed in this chapter, you would have installed and configured Linux to run your computer. You have been through the installation

process (probably more than once), laid out your file system, connected to your home or work network, and became comfortable with operating Linux from the command line. Now what? Your computer simply has an operating system. It is sitting there with little lights flashing, making whirring noises, but it has no purpose, yet.

Applications define a computer's destiny. Applications make it a server or a workstation. The operating system, be it Linux or any other, turn a computer into an application platform. Its applications transform the computer from useless waste of electricity into a useful and hopefully productive tool.

Once Linux is installed and configured, as an administrator or power user, most of your time will be spent maintaining the applications. This involves installing, configuring, supporting, and removing them. This chapter will walk you through these processes using a variety of methods and tools. The two most prevalent tools for managing application packages are advanced packaging tool (APT) and RedHat package manager (RPM). Most leading distributions use one of these tools.

When you need to install applications that are not packaged, you will need to resort to compiling and installing from the source code. Compiling code sounds like it falls in the realm of the application developer or programmer. The source code of open source software, by its very nature, is available for download, review, and installation.

> **Note**
> Learning to install from source is also useful for configuring and compiling customized Linux kernels.

INSTALL, REMOVE, AND UPDATE PROGRAMS

Within the Linux system there are a number of ways to install programs or *packages* as they are commonly known within Linux. A package can be considered to be a group of files that are bundled together into one archive file. Each of these packages can be an entire application or perhaps just a group of related *library files.* All software for Linux will come in the form of a package, and you will need to learn how to install software using these packages.

Within Linux there are a number of different package formats and the ones you will need depends entirely on the Linux distribution you are running. Learning the basics of installing all these types of packages is extremely useful and will ensure that you have a broad understanding of the mechanics of

each process. Whichever package type you need to install, each program will usually have one or more associated libraries or support packages that have to be installed with it to make it work. These additional packages are called *dependencies* as the main program is dependent on these to work. If you already have these dependencies installed, then you do not have to reinstall them. This differs to installing a program under Microsoft Windows which will have everything bundled into one file, and often multiple versions of a particular library may be loaded on the same system, just in a different location.

In principal, there are two types of packages: *binary packages* and *source packages.* The source packages will need to be compiled and built for your system while the binary packages have already been compiled for a specific installation. Obviously, the binary packages cannot easily be modified to suit any specific need you may have, but, in general, these are much easier to install, especially for a novice user. The utilities that distribute and manage the binaries for a particular distribution are called *package managers*. The various packages can be identified by their suffix as shown in Table 7.1 below.

The packages can be found individually on various Web sites or on installation disk, but also in *software repositories*. These are locations where the software packages can be found, downloaded, and installed on your computer. These repositories can vary from having a small number of packages (even one) on them, or with a whole operating system on them. For instance, the packages for Debian GNU/Linux can be found here: www.debian.org/distrib/packages.

Within this chapter, we will concentrate on the two main packages, rpm and deb and also how to compile and build the software from the source files. These will give you the grounding you need to tackle some of the other less well used formats should you need to.

Table 7.1	Linux Package Formats
.rpm	RPM package manager is used by Red Hat, openSUSE, Mandriva Linux, Mandrake, and many more.
.urpm	Extended form of rpm is used by later versions of Mandriva.
.deb	Debian package is used by Debian and Ubuntu, Knoppix.
tgz or tar.gz	tar and gzip package is used by Slackware.
Other	Lots of others, mainly for the smaller distros.

Red Hat Package Manager

RPM has a long history, and was first used back in 1995 with Red Hat Linux 2.0. RPM stands for Red Hat package manager, which is a recursive acronym. Within version 3.0 of Red Hat Linux, RPM was completely redesigned and rewritten in C, and has been a feature of this and a number of other distributions ever since. The early versions of the software were limited to the command line only, but now there are graphical user interfaces (GUIs) available for most of the distributions that use it. The basic rpm file is a precompiled binary package bundled with a script file that minimizes the knowledge the end-user needs.

The Red Hat package manager is an overall system called RPM and this can be used to build, install, remove, modify, and verify the software archives or .rpm files. The `rpm` packages contain a complete archive of the files, along with a host of other information on the package, such as name, version, and checksums. In addition, there can be scripts included to install, upgrade, or remove the software, along with preinstallation and postinstallation scripts where necessary. The checksum is particularly useful as it can validate the binary and ensure that the software is free from viruses and trojans.

The RPM system uses a database to hold and track all of this information, including the version of all the software that is installed. This basic information is held in the /var/lib/rpm directory and a sample listing of this directory is shown below:

```
$ ls -l
total 46184
-rw-r--r-- 1 root root  2994176 2009-06-23 12:29 Basenames
-rw-r--r-- 1 root root    12288 2009-06-23 12:29 Conflictname
-rw-r--r-- 1 root root        0 2009-06-23 12:13 __db.000
-rw-r--r-- 1 root root    24576 2009-06-23 12:31 __db.001
-rw-r--r-- 1 root root   180224 2009-06-23 12:31 __db.002
-rw-r--r-- 1 root root  1318912 2009-06-23 12:31 __db.003
-rw-r--r-- 1 root root   352256 2009-06-23 12:31 __db.004
-rw-r--r-- 1 root root  1507328 2009-06-23 12:29 Dirnames
-rw-r--r-- 1 root root  5300224 2009-06-23 12:29 Filedigests
-rw-r--r-- 1 root root    32768 2009-06-23 12:29 Group
-rw-r--r-- 1 root root    24576 2009-06-23 12:29 Installtid
-rw-r--r-- 1 root root    45056 2009-06-23 12:29 Name
-rw-r--r-- 1 root root 35545088 2009-06-23 12:29 Packages
-rw-r--r-- 1 root root   331776 2009-06-23 12:29 Providename
-rw-r--r-- 1 root root   118784 2009-06-23 12:29 Provideversion
-rw-r--r-- 1 root root    12288 2008-11-19 14:17 Pubkeys
```

```
-rw-r--r-- 1 root root   503808 2009-06-23 12:29 Requirename
-rw-r--r-- 1 root root   270336 2009-06-23 12:29 Requireversion
-rw-r--r-- 1 root root   163840 2009-06-23 12:29 Sha1header
-rw-r--r-- 1 root root    86016 2009-06-23 12:29 Sigmd5
-rw-r--r-- 1 root root    12288 2009-06-23 12:29 Triggername
```

The /var/lib/rpm/packages file is the primary database of all the installed software in the system and will grow depending on the number of packages you install. A file of 40 MB or larger is not an unusual size for a fully loaded system. This directory is used by RPM to manage all the software and versions.

The `rpm` package names are in a standard format and contain

- Package software name
- Version of the software
- The release number of the package
- Architecture the package was built for (for example, i386, i686, and so forth). The actual format will be as follows:

  ```
  <package_name>-<version>-<release>.<arch>.rpm
  ```

As an example, the `rpm` for a Telnet package file, with a version of 0.17 and a release of 23 built for the i386 platform would look like

```
telnetd-0.17-23.i386.rpm
```

Command Line Tools

Although the new GUI front-ends to RPM are easy, the mechanics behind the actual commands are masked and does not demonstrate the power of the `rpm` command. Both the GUI and the command line interface (CLI) are similar in one respect – they need to be executed by the superuser account. The basic operations that we will examine in the following sections are as follows:

- Installation of new software
- Removing (or erasing) packages
- Upgrading an existing package
- Verifying the installation of a package
- Information querying regarding an installed package

While the actual number of command line options is far greater than this list, they all can be grouped into one of the above-mentioned sections. The basic format of the `rpm` command is

```
rpm option package_name
```

Taking the list above, the options for these basic operations are as follows:

- `-i` will install the package.
- `-e` will remove (erase) the package.
- `-U` will remove the installed package first and then install the new version.
- `-V` will verify the installation of the package.
- `-q` will query the package.

Each of these can be combined with a number of other options to make the `rpm` command very powerful. The following sections will explore these options in more detail.

Package Installation

Packages are installed as stated above using the `-i` option. A simple install of the earlier-specified Telnet package would, therefore, be

```
rpm -i telnetd-0.17-23.i386.rpm
```

There are a number of useful options that are often combined with the `install` option, as well as with the other options, namely `-v`, `--quiet`, and `-h`. The `v` option is for verbose and will give some useful feedback during the process. The `--quiet` option is the exact opposite – displays as little information as possible. The `-h` option will print a series of hash marks on the screen as the work proceeds, with the sole purpose of keeping you aware that something is still happening.

When installing a package, you may not wish to have the documentation installed, and this is achieved using the `--excludedocs` option. Files can be forced to replace existing files using `--replacefiles` and `-force` can also be used to ensure the entire install is forced onto the system.

Package Updating

Packages already installed on system can be upgraded to a later release using the `-U` option. This option will remove the old version, keeping any modified files such as the configuration files. It will then install the new version.

To upgrade the earlier-specified Telnet package to version 18, release 5 the command to use is as follows:

```
rpm -U telnetd-0.18-5.i386.rpm
```

To see this upgrade in verbose mode and printing hash marks when the archive is unpacked, the command would look like

```
rpm -Uvh telnetd-0.18-5.i386.rpm
```

Package Querying

The query command, `-q`, will query the rpm database and give you data on the package. For instance, to find out about version of Firefox installed on the system, the following is used (showing the output):

```
$ rpm -q firefox
firefox-3-0-11-1.fc10.i386
$
```

Information on all the packages installed can easily be displayed using `rpm-qa`, but the output will be very long and should be piped through `-more` or redirected to a file.

Package Removing

Packages already installed on a system can be removed, which you may want to do to conserve space or if there is a problem with the package. Packages can be considered to be removed or erased with the `-e` option. For instance, removing the earlier-specified Telnet package can be achieved thus:

```
rpm -e telnetd-0.17-23.i386.rpm
```

If you don't want rpm to check dependencies before uninstalling the package, the following option can be added `--nodeps`. Another useful option is `--test`, which will go through the motions of the uninstall process, but will not actually delete anything.

Yellow Dog Updater Modified

There is an automatic installation, removal, and update utility for rpm packages called *yellow dog updater modified (yum)*. This was developed using the Python language, initially for the Fedora Linux distribution, and it is now part of the Fedora distribution. The latest version can be used with a GUI interface as well as the CLI. We will concentrate on the command line to completely understand the process.

Fedora Linux provides a number of software repositories, and is preconfigured to work with three repositories:

- ***Base repository*** contains the Fedora release, usually on your installation media.
- ***Updates*** will have all the updates from the base package.
- ***Extras*** is a large selection of additional software that a user may want to install.

In addition there are also development repositories available which will have the newest code, but which may not be stable. Like rpm, `yum` needs to have superuser privileges to be performed. The available package groups provided by the Fedora repositories can be queried using `yum` using the command `su -c 'yum grouplist'` and entering the superuser password when prompted. Part of the output can be seen below:

```
$ su -c 'yum grouplist'
Password:
Loaded plugins: refresh-packagekit
Setting up Group Process
Installed Groups:
  Administration Tools
  Base
  Dial-up Networking Support
  Engineering and Scientific
  Fonts
  GNOME Desktop Environment
  Games and Entertainment
  Graphical Internet
  Graphics
  Hardware Support
  Input Methods
```

> **Note**
> As the root user, you have extreme power and can add and delete files, and mistakes can be made. Managing a Linux system can be achieved without logging in as root, but running them using a substitute user or switch user command (`su` or `sudo`) depending on the Linux distribution you are using.

Installing Software with Yum

You can install new software packages or package groups with yum. The following show the commands for a single package (Firefox) and a package group (MySQL database):

```
su -c 'yum install firefox' su -c 'yum
groupinstall 'MySQL Database'
```

The sample output to install a simple package (tsclient) is shown below:

```
$ su -c 'yum install tsclient'
Password:
Loaded plugins: refresh-packagekit
Setting up Install Process
Resolving Dependencies
--> Running transaction check
---> Package tsclient.i386 0:2.0.1-2.fc10 set to be updated
--> Processing Dependency: rdesktop >= 1.3.0 for package:tsclient-
2.0.1-2.fc10.i386
--> Processing Dependency: vnc >= 4.0 for package: tsclient-2.0.1-2.fc10.i386
--> Running transaction check
---> Package rdesktop.i386 0:1.6.0-2.fc10 set to be updated
---> Package vnc.i386 0:4.1.3-1.fc10 set to be updated
--> Processing Dependency: librfb.so.0 for package: vnc-4.1.3-1.fc10.i386
--> Running transaction check
---> Package vnc-libs.i386 0:4.1.3-1.fc10 set to be updated
--> Finished Dependency Resolution

Dependencies Resolved
============================================================================
Package                Arch         Version         Repository         Size
============================================================================


Installing:
tsclient               i386         2.0.1-2.fc10    fedora             106 k
Installing for depen-
dencies: rdesktop      i386         1.6.0-2.fc10    fedora             147 k
vnc                    i386         4.1.3-1.fc10    updates             90 k
vnc-libs               i386         4.1.3-1.fc10    updates            167 k

Transaction Summary
============================================================================
Install       4 Package(s)
Update        0 Package(s)
Remove        0 Package(s)

Total download size: 509 k
Is this ok [y/N]: y
Downloading Packages:
(1/4): rdesktop-1.6.0-2.fc10.i386.rpm                |  147 kB    00:00
```

```
(2/4): tsclient-2.0.1-2.fc10.i386.rpm       | 106 kB    00:00
(3/4): vnc-4.1.3-1.fc10.i386.rpm            |  90 kB    00:00
(4/4): vnc-libs-4.1.3-1.fc10.i386.rpm       | 167 kB    00:00
-------------------------------------------------------------
Total                              225 kB/s  | 509 kB    00:02
Running rpm_check_debug
Running Transaction Test
Finished Transaction Test
Transaction Test Succeeded
Running Transaction
   Installing         : vnc-libs-4.1.3-1.fc10.i386      1/4
   Installing         : vnc-4.1.3-1.fc10.i386           2/4
   Installing         : rdesktop-1.6.0-2.fc10.i386      3/4
   Installing         : tsclient-2.0.1-2.fc10.i386      4/4

Installed:
   tsclient.i386 0:2.0.1-2.fc10

Dependency Installed:
   rdesktop.i386 0:1.6.0-2.fc10    vnc.i386 0:4.1.3-1.fc10
   vnc-libs.i386 0:4.1.3-1.fc10
Complete!
```

As you can see, the installation of *tsclient* started by checking for packages that are needed for this installation, or rather the dependencies for this package. This leads to two dependencies: *VNC* and *rdesktop*, which in turn also needed an extra library. The dependency check needs to be recursive, and so new packages that have to be installed are checked for dependencies and so on. For a large complicated package (for example, to install the KDE or GNOME environment), this may lead to a large number of dependencies.

Exam Warning

When you install a service, the Linux system will not start it. You must configure the service to run on boot up, which can be achieved from the command line using `chkconfig` and `service` commands.

Updating Software

To update a software package that is already installed, you can use the `update` option within yum. For example, to update the tsclient package

you would type the command `su -c 'yum update tsclient'`. If the software is not installed, the system informs the user as shown below:

```
$ su -c 'yum update tsclient'
Password:
Loaded plugins: refresh-packagekit
Setting up Update Process
Package(s) tsclient available, but not installed.
No Packages marked for Update
```

If the software being updated is currently in use by the system, the application or service will need to be restarted before the update is made current. The kernel can also be updated using `yum` and these updates will only come into force upon a restart of the system. When the kernel is updated, `yum` will retain the old version so that the old kernel can be booted into in case of an error with the new kernel. Only the current and previous versions are kept.

Package groups can also be updated, for example, the MySQL Database package group is updated using the command `su -c yum groupupdate 'MySQL Database'`.

Removing Software

The removal of software is achieved using the `remove` option. Again, both packages and package groups can be removed. When this is invoked, yum will check the software package and the dependencies and will remove both.

```
su -c 'yum remove firefox' su -c 'yum
groupremove 'MySQL Database'
```

When yum removes the software package, it leaves any user data in place, but the configuration files may be removed.

deb

The Debian-derived distributions of Linux are based on the GNU project, which is a project that started in 1984 to produce an open-source Unix-like operating system. While it is often referred to as Linux, the correct name is Debian GNU/Linux as it combines both. The current version of Debian includes more than 25,000 packages. The Debian packages can be considered to be similar to the rpm packages in that they are precompiled for easy installation on the target system.

There are three main package libraries available from the Debian package Web site (http://packages.debian.org):

- **Stable libraries** are well tested and will change only for security or major bug fixes.

- **Testing libraries** have had a lot of testing and are destined to be in the next release.
- **Unstable** have had little testing and may well contain bugs that could make the system unstable.

The low level or base tool of the Debian package manager is the command `dpkg`. This command and the main options will be discussed below. In addition to this tool, there are also a number of higher-level tools such as APT, which can be used to fetch packages from many locations. Also, there is a tool called `aptitude` that has a much easier and friendly-user interface.

Installing Software Packages using `dpkg`

The basic install of packages for using the Debian package manager is very similar to using RPM in the previous section. To install a package the following is used:

```
dpkg -i package_name
```

As with rpm above, you will need to have superuser privileges to run the command. Without this, the following error message will be displayed:

```
dpkg: requested operation requires superuser privilege
```

Removing Software Packages using `dpkg`

Packages can be removed easily using the `-r` option:

```
dpkg -r package_name
```

This option will leave the configuration files on the computer so that it will be easier to reinstall it later. If you want to erase the configuration files as well, you can add the `--purge` option.

Advanced Packaging Tool

Another package management tool for Debian is *APT* or *Advanced Packaging Tool*, which was designed to facilitate the administration of the packages. The default location for the APT configuration files is /etc/apt. APT has a list of locations where packages can be obtained from and this is in /etc/apt/sources.list, part of which is shown below:

```
#
# deb cdrom:[Debian GNU/Linux 5.0.1 _Lenny_ - Official i386 DVD Binary-1
20090413-00:33]/ lenny contrib main

deb cdrom:[Debian GNU/Linux 5.0.1 _Lenny_ - Official i386 DVD Binary-1
20090413-00:33]/ lenny contrib main
```

```
deb http://security.debian.org/ lenny/updates main contrib

deb-src http://security.debian.org/ lenny/updates main contrib

deb http://volatile.debian.org/debian-volatile lenny/volatile main contrib

deb-src http://volatile.debian.org/debian-volatile lenny/volatile main contrib
```

There is an internal database kept by APT to track the packages that are currently installed, those that are not installed, and those that are available to be installed. You can use the `apt-get` command to query this database, install and remove packages, and to check for dependencies in packages. As this list changes when new packages are added and new dependencies come into force, the list needs to be updated. This is achieved using the following command:

```
apt-get update
```

Installing Packages

Packages can be installed using the `install` option, with the general syntax of

```
apt-get install package_name(s)
```

An example of the first part of the output is shown below when the `abiword` package is installed:

```
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  abiword-common abiword-help abiword-plugin-grammar abiword-plugin-mathview
  aspell-en doc-base latex-xft-fonts libaiksaurus-1.2-0c2a
  libaiksaurus-1.2-data libaiksaurusgtk-1.2-0c2a libfreezethaw-perl
  libfribidi0 libgdome2-0 libgdome2-cpp-smart0c2a libgoffice-0-4
  libgoffice-0-common libgsf-gnome-1-114 libgtkmathview0c2a liblink-grammar4
  libloudmouth1-0 libmldbm-perl libots0 libt1-5 libuuid-perl libwv-1.2-3
  link-grammar-dictionaries-en
Suggested packages:
  abiword-plugin-goffice
The following NEW packages will be installed:
  abiword abiword-common abiword-help abiword-plugin-grammar
  abiword-plugin-mathview aspell-en doc-base latex-xft-fonts
  libaiksaurus-1.2-0c2a libaiksaurus-1.2-data libaiksaurusgtk-1.2-0c2a
  libfreezethaw-perl libfribidi0 libgdome2-0 libgdome2-cpp-smart0c2a
  libgoffice-0-4 libgoffice-0-common libgsf-gnome-1-114 libgtkmathview0c2a
```

```
  liblink-grammar4 libloudmouth1-0 libmldbm-perl libots0 libt1-5 libuuid-perl
  libwv-1.2-3 link-grammar-dictionaries-en
0 upgraded, 27 newly installed, 0 to remove and 0 not upgraded.
Need to get 0B/9858kB of archives.
After this operation, 31.3MB of additional disk space will be used.
Do you want to continue [Y/n]? y
Selecting previously deselected package libaiksaurus-1.2-data.
(Reading database ... 95088 files and directories currently installed.)
Unpacking libaiksaurus-1.2-data (from .../libaiksaurus-1.2-data_1.2.1+dev-0.12-
6_all.deb) ...
```

This will invoke `apt-get` to search the database for the most recent version of the package and then will retrieve it from the location specified in /etc/apt/sources.list and, if there are any dependencies, install these packages as well. The option `-y` can be used with the `install` option to assume *Yes* to all queries to reduce the user interaction. You can also reinstall a package if you want to ensure that the files are the newest available or if you suspect some have become corrupted.

```
apt-get --reinstall install package_name(s)
```

You can download the packages for later installation using the `-d` option with the *install* option. These files are stored in the /var/cache/apt/archives directory.

```
apt-get -d install package_name(s)
```

Package Removal

The packages are removed from the system using the `remove` option.

```
apt-get remove package_name(s)
```

The package will be removed, along with all the packages that depend on it. The configuration files will not be removed, but adding the option `--purge` will remove all files associated with the package. Using the `purge` option is worthwhile if you know you will not be using this package in the future, as it will clean up the disk and not leave a lot of unwanted files in the filesystem.

Advanced Installation and Removal

You can install and remove packages on the same command line. For instance, to remove a package (or packages) when you are installing a package by suffixing them with a "-", the syntax of the command would look like

```
apt-get install newpackage oldpackage-
```

Hence to install python package and remove the nano package, use

```
apt-get install python nano-
```

Conversely, to install a package while removing one is achieved by suffixing the package to be installed with a "+"

```
apt-get remove oldpackage newpackage+
```

In the above example, we could remove the nano package and install python using

```
apt-get remove nano python+
```

Upgrading Packages

The upgrading of packages can be accomplished very easily within the APT system. All the packages within the current distribution can be upgraded with a single command

```
apt-get upgrade
```

For upgrading the packages to a new distribution, it is better to use the command below to ensure that all relationships between packages are updated:

```
apt-get dist-upgrade
```

For both commands, it is worth adding the option `-u` to ensure that there is sufficient output for you to see what is being upgraded. The initial part of the output is shown below:

```
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages will be upgraded:
  libcupsimage2 libcupsys2 libdns45 libebook1.2-9 libecal1.2-7
  libedata-book1.2-2 libedata-cal1.2-6 libedataserver1.2-9
  libedataserverui1.2-8 libegroupwise1.2-13 libexchange-storage1.2-3
  libfreetype6 libgdata-google1.2-1 libgdata1.2-1 libglib2.0-0 libicu38
  libisc45 libisccc40 libisccfg40 libkrb53 liblwres40 libmozjs1d
  libmysqlclient15off libnss3-1d libpango1.0-0 libpango1.0-common
  libpoppler-glib3 libpoppler3 libpostproc51 libpurple0 libsasl2-2
  libsasl2-modules libsmbclient libssl0.9.8 libvolume-id0 libwbclient0
```

Learn by Example: Getting Out of "Dependency Hell"

When I first started using Linux in the late 1990s, I selected a particularly well-known distribution for no other reason than the fact that it was popular and that I would not have to create a bunch of floppy disks to start the installation. I suppose that I was a bit impatient. In my own process of learning about Linux, I progressed to the point where I was installing a variety of applications and my chosen distribution used RPM as the

package manager. Before long, I found myself in a very frustrating place, an unfriendly place called "Dependency Hell." I would download an .rpm package and issue the `rpm -Uvh` command, only to discover that I needed one or two or five more .rpm packages because the application I wanted to install needed these other files to be installed in order to run. I would chase down these other packages, only to discover that often I would need additional .rpm packages because the dependencies had dependencies. A seemingly simple installation that would eventually take minutes to complete actually took an hour or more because I had to hunt down, download, and install all of the dependencies. For this reason, I switched to a system that used APT as a package manager. APT does its own dependency checking and if there are dependencies, it will consult a list of application repositories that are scattered around the Internet and automatically download and install them along with my desired application. It was like I died and went to Heaven.

Obtaining Information About Packages

You may often want to install a package but are not sure what the name of the package is. One method of finding these packages is to use `apt-cache`. To search for packages you will need to use

```
apt-cache search name
```

For instance, suppose you want to search for *abiword*, you would execute the command `apt-cache search abiword`, with the output shown below:

```
abiword - efficient, featureful word processor with collaboration
abiword-common - efficient, featureful word processor with
collaboration -- common files
abiword-help - online help for AbiWord
libgtkmathview0c2a - rendering engine for MathML documents
abiword-plugin-grammar - grammar checking plugin for AbiWord
abiword-plugin-mathview - equation editor plugin for AbiWord
abiword-plugin-goffice - GOffice interaction plugin for AbiWord
```

To gain further information on a particular package, you will need the following command `apt-cache show abiword`, and the first part of the output is shown below (edited for brevity):

```
Package: abiword
Priority: optional
Section: editors
Installed-Size: 7352
Maintainer: Masayuki Hatta (mhatta) <mhatta@debian.org>
Architecture: i386
Version: 2.6.4-5
```

```
Replaces: abiword-gnome
Provides: abiword-gnome
Depends: libaiksaurus-1.2-0c2a (>= 1.2.1+dev-0.12), libaiksaurusgtk-1.2-0c2a
(>=1.2.1+dev-0.12), libart-2.0-2 (>= 2.3.18), libatk1.0-0 (>=1.20.0), libc6 (>=
2.7-1), libcairo2 (>= 1.2.4), libenchant1c2a, libexpat1 (>= 1.95.8), libfontconfig1
(>=2.4.0), libfreetype6
Recommends: abiword-plugin-grammar, abiword-plugin-mathview, abiword-help,
aspell-en | aspell-dictionary, poppler-utils
Suggests: abiword-plugin-goffice
Conflicts: abiword-gnome
Filename: pool/main/a/abiword/abiword_2.6.4-5_i386.deb
Size: 2882324
MD5Sum: 7fabfdf5ea014d67541441b930674ff0
SHA1: 792d8d83177ef23cc802b7c249b47b12fa797031
SHA256: c642cd84e17d9e0e88c539f10e812ea187d8a2861acc2566ff1ca21a5
55ead3d
Description: efficient, featureful word processor with collaboration
 AbiWord is a full-featured, efficient word processing application.
 It is suitable for a wide variety of word processing tasks, and
 is extensible with a variety of plugins.
 .
 This package includes many of the available import/export plugins allowing
 AbiWord to interact with ODT, WordPerfect, and other formats. It also
 includes tools plugins, offering live collaboration with AbiWord users
 on Linux and Windows (using TCP or Jabber/XMPP), web translation and
 dictionary support, and more.
 .
 Additional plugins that require significant amounts of extra software to
 function are in the various abiword-plugin-* packages.
Tag: interface::x11, role::program, scope::application, uitoolkit::gtk,
use::editing, use::text-formatting, works-with::text, works-with-
format::html, works-
with-format::tex, x11::application
```

Compiling and Installing Applications from Source

From experience, compiling applications from source is something you will inevitably run into in your career as a Linux administrator. Compiling and installing from source presents a terrific opportunity to tune applications to your specific hardware and software platform. This section starts with a description of how and where to include these hardware- and software-specific parameters and then continues through the process to compile and install the application. It concludes with several prominent utilities for archiving and packaging source code: `tar`, `bzip`, and `gzip`.

Configuring the Source

We will discuss downloading the package archive later on in the section, but suppose for now that you have the source in a suitable directory on your hard drive. The first place to look is to see if there are any README or INSTALL files in the directory – and if they are found, read them. These usually contain very useful information on the software and often will give details on how to install them with any specific options or dependencies that you may need.

After reading the documentation, you need to change directories to the directory where the package is stored. You can then configure the package, which is usually achieved using the `configure` script by typing **`./configure`**.

> **Note**
> It is very likely that the directory containing the new package will not be in your current path, and hence you need to be specific by using `./configure` and not just `configure`.

The `configure` script is a shell script which will configure the *makefile*, which is used by the compilation tool *make* (described below). This *makefile* will have information on your system to enable *make* to compile the source correctly. Originally, the *makefile* had to be edited by hand which, from experience, could take a long time. The machine output from this command will be a new *makefile*, and if the command worked correctly, this will be constructed and placed in the correct directory. There could be a lot of messages scrolled to the standard output (often the screen) during this process. If it finds an error, it will be reported and configure will exit. If there are no errors, configure will end gracefully.

make

The utility `make` is used to automatically determine which components of a software package need to be complied, and then to guide this compile process. The utility can be used with any type of software package that can be compiled with a shell script. The `make` utility will use the *makefile*, which details the relationships among the files in the package and how to update these files. This will be undertaken from the data in the database and the last modification times of the files. The executables are typically made up from *object files*, which themselves are compiled from *source files*.

Once you have the *makefile* after running *compile*, then you can run `make`. This shell script is typically run initially with no option. This will parse the *makefile* and update any files as necessary. If `make` completes, this

will build a binary of the software package. This does not install the binary; that step is achieved in the last step using the command

```
make install
```

If this exits with no errors displayed, then the software has been installed correctly. This will have to be run with superuser privileges. The `configure` script will determine where the program will be installed, typically in */usr/local/bin*. To clean up your system from the temporary files left by `make`, you can run the command `make clean`. Unlike one of the package managers described in previous sections, it is often not as easy to remove programs installed by this process. If the *makefile* is still there, you might try to use the command `make uninstall`, but this often does not work. At this point, you will have to uninstall the programs manually.

autoconf

The `autoconf` utility is a package of M4 macros that are used to build a set of shell scripts to automatically configure software source packages. The utility will create a configuration script from a template file listing the operating system features that the package uses. The generation of the configuration files is primarily to make the user's experience easier, to ensure the configure process is easier to use and less prone to errors.

Archive Files

The *tar* file format has been in existence since the early days on UNIX and was originally designed for *t*ape *ar*chives (*tar).* The utility to use this file format is also called `tar`. It is now a common method for archiving or collecting a large number of files into one larger file, while preserving all the file information, such as directory structures, dates, and user and group permissions. Files that are packed into this format have a naming structure of *filename.*`tar`. These large files can be compressed using a compression utility such as `gzip`, `bzip`, or `compress`. Depending on the compression utility used, the tar file will be renamed.

- *filename.tar* will become *filename.tar.gz* if `gzip` is used.
- *filename.tar* will become *filename.tar.bz* or *filename.tar.bz* 2 if `bzip/bzip2` is used.

Before we look at the compression utilities, we will look at the `tar` utility. This is often used within Linux and the syntax and operation should be known to any Linux administrator. To create a tar file, the following general syntax can be used:

```
tar -cvf filename.tar files|directories
```

The options used are `c` to create a tar file; `v` for verbose output; and `f` to put the output in the specified file. The tar archive will be created from one or more files and/or directories specified at the end. There could be multiple files and directories specified on the same command line. For instance, suppose you wanted to compress everything in your work directory in your home folder, say /home/syngress/work. The command to create an archive work.tar in the current directory would be as follows:

```
tar -cvf work.tar /home/syngress/work
```

Once the tar file has been created, you can list its contents by using `tar -tvf work.tar`. The tar file can be decompressed or the files extracted using `tar -xvf work.tar`. This extraction process does not remove the tar file, but places the files in the current working directory.

The `tar` utility can also be used to compress the tar file that has been created. In the above example, to compress the tar file of the work directory you would use

```
tar -czvf work.tar
```

The files are compressed using `gzip` and will be given the *.tgz* extension. The compressed file can be decompressed by using

```
tar -xzvf work.tar
```

Compression Utilities

There are a lot of compression utilities available, and it is often a personal choice which one to use. Some utilities work best on certain types of file, but we will concentrate on a couple of them. We have just mentioned `gzip` which can be used with `tar`. It is also a stand-alone program that can be invoked to compress files at any time. The format of the command is as follows:

```
gzip filename.ext
```

This will compress the file *filename.ext* and save it as *filename.ext.gz*. The original file will be deleted during the process. This can be decompressed by using the following command:

```
gunzip filename.ext.gz
```

Again, the command will delete *filename.ext.gz* and leave *filename.ext* only. The `gzip` utility can compress files to different levels from 1 through 9, with 1 being quick but least efficient, to 9 being slow but very efficient. The default is a level 6, which could be increased due to the speed of modern computers, but often the newest computers have the most disk space! For both `gzip` and `gunzip` the option `-r` can be used which will recursively compress or decompress all the files in the current directory and the subdirectories.

The `bzip2/bunzip2` utilites are another pair of compression and decompression tools, which often give slightly better compression ratios. The command line options are very similar to that of `gzip/gunzip`. The compressed files will usually have an extension of bz/bz2 or tbz/tbz2 (compressed file or a compressed tar file).

EXERCISE 7.1: Installing Software from Source Code

You need to install postgreSQL on your system instead of MySQL by using the source. You have the gzip source file on your disk. The following will unpack, configure, and install the software:

1. Extract the software from the archived file using the following command:

   ```
   tar xvfz postgresql-8.1.3.tar.gz
   ```

2. Change directories into the topmost directory you have just created.

   ```
   cd postgresql-8.1.3
   ```

3. Configure the *makefiles* for *make.*

   ```
   ./configure
   ```

4. Compile the sources by typing `make`.
5. Install the software using the following command:

   ```
   make install
   ```

You have now successfully extracted, configured, and compiled the postgreSQL software package. ■

RESOLVING APPLICATION DEPENDENCIES

As you may have read in the sidebar earlier in the "Install, Remove, and Update Programs" section of this chapter, you can download dependent application packages manually as you try to install an application from an .rpm package, or you can use `yum`. Yum's full name is "Yellow dog Updater, Modified." Yellow Dog is a reference to Yellow Dog Linux, a Linux distribution that focused on putting Linux on Apple hardware. Yum is used on RPM-based systems, such as Red Hat, Fedora, CentOS, and Yellow Dog, itself. In addition, the application repositories in openSUSE are exclusively Yum-based.

For Linux distributions that use `dpkg`, APT natively resolves application dependencies. Although APT is a terrific tool from the command line, you are not forced to open a terminal window or change runlevels to drop to a command line when working in a GUI environment. Aptitude and Synaptic are often installed by default dpkg-based systems where X and a window manager are installed. If not, they can be easily installed from the command line using APT.

EXERCISE 7.2: Resolving Dependencies Using Yum

You know that a new version of `abiword` has been released, along with a number of new libraries. You do not want to install each of these individually, so you decide to use yum to resolve these dependencies.

1. To check the version of `abiword` and what dependencies are required, you need to type

   ```
   yum update abiword
   ```

2. The current version of `abiword` will be shown and what dependencies need to be updated and/or installed.
3. The system will then ask if you want these upgraded and installed. Type **Y**.
4. The system will upgrade and install all the software, and respond with `complete.`
5. You have now successfully resolved all the dependencies for `abiword` and installed everything correctly. ■

ADDING AND REMOVING REPOSITORIES

There are thousands of software packages for Linux, no matter what distribution you are running. These packages are stored in software repositories, and the main repositories for the particular distribution you installed are usually set up at that time. If you are not connected to the Internet during the initial load of Linux, these may not be set up, or marked as inactive.

Yum Repositories

Software repositories can be defined on remote servers as well as locally. The repositories are defined in the /etc/yum.conf file and in /etc/yum.repos.d directory. You can make a local repository by downloading the software from other

repositories, and then setting up a local repository to save downloading these for a number of machines on your network. When you have downloaded the packages, you need to generate the correct information for a repository. This is achieved using the `createrepo` utility, which extracts all the data from the rpm files to generate the necessary metadata for `yum`. The command to create the metadata from the rpm files in the /rpm_directory is as follows:

```
Createrepo /rpm_directory
```

This can then be included into its own file in the /etc/yum.repo.d directory.

Adding a Repository in Debian

The method of adding a repository that uses the APT packaging tool is different. The file /etc/apt/sources.list contains a list of available software repositories, and it can be updated manually or (if installed) by a graphical manager tool. If you want to add a new repository manually, the format to follow is: package type, Web address (URL), distribution, and section. For example, one of the lines in sources.list could be

```
deb http://security.debian.org/ lenny/updates main contrib
```

EXERCISE 7.3: Adding a New Software Repository

You want to add a repository you know exists at ftp://ftp.nerim.net/debian-marillat/. This is achieved by

1. Opening the file /etc/apt/sources.list using your favorite editor, such as `vi`.
2. Adding in the line:

   ```
   deb ftp://ftp.nerim.net/debian-marillat/ etch main
   ```

3. Save and exit from the editor.
4. You have now added the new repository to your system. ■

SUMMARY OF EXAM OBJECTIVES

Within this chapter, you learned how to download and install applications using a variety of methods, both from binary packages and using source code. Initially, we looked at the software package formats that you will most likely encounter on the Linux systems you are administrating – RPM and DEB packages. The RPM format of software packages was developed by Red Hat

and is found on Red Hat, SUSE, Fedora, Centos, and many other distributions. The DEB packages are found on distributions based on the Debian code and include Debian (of course), Knoppix, Ubuntu, and a variety of other distributions. The popularity of these formats along with the support that is available for some of the major distributions will push most companies to use these.

With both the software package formats, you were guided through the main aspects of software management, namely adding, deleting, and updating the various packages. While both command sets are very similar to each other, there are a number of differences which you should understand and remember for the test. The update commands for both should be memorized and any additional options that can be added to these commands understood.

The downloading, compiling, and building software from source was described and some background on when and why this may be necessary. In particular, you may want to make modifications to the actual source code to make it more compatible to your particular Linux build. The usual method is to use `compile`, `make`, and `make install`. This sequence will generate a *makefile* using `compile`, and then `make` will run this *makefile* to compile the source. Finally, the software can be installed using `make install`. The instructions for undertaking this and any command line switches that may be necessary are usually found in the INSTALL text file located in the directory where the software is.

Linux, unlike Microsoft Windows, does not add all the libraries and other dependent packages into one large file to install on your system. Instead, the software has a number of dependencies where the installation of one package will depend on another package to be installed. The resolving of dependencies used to be time consuming and frustrating as one dependency leads to another and another. This has been solved using a number of tools such as `yum` and `apt` that will work out what are all the dependencies and install these as well. This will reduce your workload considerably, and usually means you just have to remember the syntax of the high level tool, and let it work out what to do.

All of these packages are located in software repositories, which can be considered to be buckets containing one or more software packages. Once these are defined on your system, when you want to download a new software package, the system will look into these repositories to download the package and install it. In addition, updates to the system will be located in the software repositories and the system will use these to compare with the version of software you have loaded and suggest that any are upgraded if a newer version is released.

SELF TEST

1. A user in your finance department has approached you to let you know that an update is available for one of their core applications. It is critical that the installation go as smoothly as possible. You have been asked to perform the upgrade. What is the correct syntax for safely upgrading the existing application?

A. `rpm -uvh`

B. `rpm -ivf`

C. `rpm -Uvh`

D. `rpm -Ivh`

2. You have used the command `tar -czvf work.tar` to compress a tarball. What will the result be?

A. An archive file compressed with `gzip` and given the `gz` extension

B. An archive file compressed with `gzip` and given the `tgz` extension

C. An archive file compressed with `bzip` and given the `bz2` extension

D. An archive file compressed with `bzip2` and given the `bz2` extension

3. You have downloaded the source files for a program you want to install. You have unpacked the archive and want to see if there are any instructions on how to compile it. What should you look for first?

A. Look for a file called `configure` in the directory structure.

B. Look for a file called INSTALL in the directory structure.

C. Look for a file called FIRST in the directory structure.

D. Look for a file called `make` in the directory structure.

4. You need to add a new local repository to a system that has .deb software repositories. Which file should you edit to achieve this?

A. /etc/apt/source.list

B. /etc/apt.d/sources.list

C. /etc/apt/apt.d/sources.list

D. /etc/apt/sources.list

5. You want to install a new ftp program onto your desktop. You are currently running version 5.4 release 8 of the software. Which file should you download to upgrade this software to the latest version available?

A. ftp-6.0-8.i386.rpm

B. ftp-9-6.0.i386.rpm

C. ftp-5.4-8.i386.rpm
D. ftp-6.0.i386.rpm

6. You have a version of Linux that is managing the software packages using *yum*. You are going to remove the MySQL database group that is currently loaded on it so you can install postgreSQL. What is the correct command to use to remove the MySQL database group?
   A. `yum removegroup MySQL`
   B. `yum groupremove 'MySQL database'`
   C. `yum remove 'MySQL database'`
   D. `yum remove --force 'MySQL database'`

7. You need to compress a series of files as much as possible as you want to put them onto a CD-ROM to send to someone and they are currently much bigger. What option would you use with gzip to achieve this?
   A. `-1`
   B. `-best`
   C. `--9`
   D. `--best`

8. One of your servers has had a drive failure and you need to restore the data from last night's backup, which is a compressed tar archive. You generated the archive of everyone's home directories with the commands:

```
cd/home
tar czvf work.tgz home
```

   You have copied the file to /tmp on the new drive and executed the command

```
tar xzvf work.tgz
```

   from that directory. To what location would the home directories be restored?
   A. They would be restored at the original location (/home) on the new drive.
   B. They would be restored to the root directory (/).
   C. They would be restored to /tmp/home.
   D. The *o* option needs to be specified to overwrite the default home directories setup by Linux or the tar command will return an error.

9. You are installing a number of new packages to an older machine that does not have a large amount of disk space and you do not want to install any documentation on the packages as you can look at this on another machine. What option should you add to rpm to ensure this happens?

   A. Include the option `--minimumsize`

   B. Include the option `--excludedocs`

   C. Include the option `--excludedocuments`

   D. Include the option `--nodocs`

10. You have downloaded the source code for a new Web program into /tmp and have read the INSTALL file that came with it. The INSTALL file says you have to run the command

    ```
    ./configure --prefix=targetdirectory
    ```

    Where do you think the *makefile* will be created?

    A. In the current directory

    B. In a subdirectory called targetdirectory from the current directory

    C. In a directory called targetdirectory in your home directory

    D. Nowhere as the *makefile* is created by `make`

11. You have downloaded the rpm files for a new program. Assuming you are a normal user, what else must you do to ensure that you can install the programs?

    A. Change the owner of the files to everyone, and run `rpm`.

    B. Run `chmod 777` on the files before executing the `rpm` command.

    C. Use the command `rpm -c rpmfile` to ensure that the system prompts you for the superuser password.

    D. Run `rpm` as superuser.

12. You want to set up a local repository on a server in your network. You are using yum and want to ensure that the repositories will work with this. What tool should you use and where should the metadata files be stored?

    A. Use `createrepo` and store the metadata in /etc/yum.repo.d

    B. Use `create-repo` and store the metadata in /etc/yum.repo.d

    C. Use `createrepo` and store the metadata in /etc/yum/yum.repo.d

    D. Use `createrepo` and the metadata will be stored automatically in the correct location

13. Which of the following commands will not upgrade an installed .deb package?
    - **A.** `apt-get install package_name`
    - **B.** `dpkg -i package_name`
    - **C.** `apt-get --reinstall install package_name`
    - **D.** `apt-get update package_name`
14. You are compiling source code for installation and you want to string all of the required commands together to run while you are going downstairs to grab a coffee so that the binary file is ready when you return. What answer below has syntax that will not work?
    - **A.** `./configure; make; make install`
    - **B.** `./configure / make / make install`
    - **C.** `./configure && make && make install`
    - **D.** `./configure | make | make install`
15. You are powering up a laptop that has Linux installed that has not been used in a couple of months. Since you will be handing it over to a user who needs to use it on a long trip, you want to ensure that its applications are current. What command do you run once the APT database is up-to-date?
    - **A.** `apt-get update`
    - **B.** `apt-get upgrade`
    - **C.** `apt-get dist-upgrade`
    - **D.** `apt-get install`

SELF TEST QUICK ANSWER KEY

1. C
2. B
3. B
4. D
5. A
6. B
7. D
8. C

9. B
10. B
11. D
12. A
13. D
14. B
15. B

This page intentionally left blank

CHAPTER 8

Installing, Configuring as a Workstation

Exam objectives in this chapter

- Printing
- X11

UNIQUE TERMS AND DEFINITIONS

- **Common UNIX printing system (CUPS)** The standards-based, open source printing system developed by Apple Inc. for Mac OS X and other UNIX-like operating systems.
- **X Window System (or simply, X)** An open source suite of software (including a network protocol) implements the X display protocol, provides windowing and manages keyboard and mouse functions to provide a graphical user interface (GUI) for networked computers.

INTRODUCTION

Users of any system, be that Linux, Microsoft Windows, Apple Mac, or any other system, will want to undertake a number of basic tasks: interaction with applications on a monitor (for example, word processing, task scheduling) and printing. Without these basics, the normal user would be at a loss and the support technicians' help calls would consume all of their time. A standard setup for the user interface and the local (and possibly remote) printers

will make the users' experience much more fulfilling as well as reduce the support overhead.

Even with the push for the paperless office, printing is still a major requirement for users, either on their home network or in the corporate environment. The installation of Linux will typically recognize and install any local printers attached to the system. Networked or remote printers will require to be installed after installation. Depending on the initial installation options, this may require additional drivers to be downloaded. The section on printers will describe how common UNIX printing system (CUPS) is configured and its use.

Most of the latest Linux distributions offer one or more graphical user interfaces (GUIs) to interact with the user. These are commonly run on top of the X Windows system, with the most common Windows managers being KDE and GNOME. We will look at the history of X up to its current incarnation, commonly called. X11 *X11*. X11 is a true client-server application, and how this is implemented is described, along with descriptions of the main configuration files that are needed.

PRINTING

Although printing is essentially a task that can be accomplished by most of the people, most of the time, the management of the devices and the setup can be confusing. The majority of printers that can be purchased today will have a Microsoft Windows driver available within the OS or have a driver disk in the box. The support for Linux drivers varies from manufacturer to manufacturer, and also the open source community provides drivers for a large number of printers. However, not all printers have a compatible driver, and users should check the availability of drivers before the purchase of a new printer.

Printing within Linux has evolved from the early days when text files could be sent to a slow parallel or serial printer connected to a server or local workstation. There are many different methods to format the output of a job to make it ready for printing, and a number of printer services. CUPS is one such print service program that is commonly deployed in a modern Linux distribution.

CUPS Overview

CUPS is developed and maintained by Apple Inc. to provide a standard printing solution, and can be deployed across a wide range of platforms. The portability of CUPS allows users to print in various environments with the

same basic display and/or commands. The CUPS application converts the page descriptions from an application into a format that the printer will understand, and then manages the process for sending this to the actual printer. Manufacturers of printers will develop different methods to print, even within their own line of printers. The CUPS application will perform this conversion, hopefully hidden from the end user.

CUPS will create and manage a queue for each printer, either locally or one accessible across the network. These queues will look the same, and the actual technicalities of where and how to send the data to the printer is handled by CUPS. Every time a user prints something, CUPS creates a job and puts it into the queue. This will include a job number to allow the user to pause or cancel a job, if required. Each of these jobs is assessed by CUPS, which then assigns the best program to convert the pages into a printable format before actually printing them. Jobs in the queue are normally processed in a first-in/first-out manner, although users can move jobs up and down the queue on occasions. Completed jobs are removed from the queue by CUPS.

With newer Linux distributions, CUPS will usually be preinstalled at installation time as part of the base system. The most recently updated version of CUPS can be downloaded from the CUPS open-source Web site http://cups.org, and is available for Linux and Microsoft Windows. In addition, the Web site provides a number of printer drivers for a range of common printers. The installation of CUPS will be covered in Chapter 9, "Installing, Configuring as a Server," and the user commands will be discussed in this chapter.

Enable and Disable Queues

When CUPS has been installed and one or more printers are set up, each printer will have its own queue which it will manage to ensure that jobs are printed in a sequential order. These queues can be enabled or disabled, even if there are items in the queue. This is often necessary when a printer has a problem that needs to be fixed and the system administrator wants to disable the queue until the issue is corrected. The following command will disable a queue called *CANONMX*.

```
# cupsdisable CANONMX
```

The printer queue can be enabled using the following command:

```
# cupsenable CANONMX
```

When a system administrator stops a printer, they can also issue a comment to tell users why the printer is disabled. This is very useful in larger corporations or when the printer is in remote locations.

```
# cupsdisable -r "Printer maintenance being performed" CANONMX
```

Web Management Port (port 631)

The interface to CUPS is through a Web interface that allows you to view print jobs and what printers are installed, and also allows for the management of these processes. This allows for easier management than using the command-line interface. Once CUPS has been set up, it can be accessed through port 631, either locally or remotely. Remote access will, of course, have to be set up, and appropriate rules are allowed in firewalls or the iptables, if used. On the local machine, the interface can be accessed using the URL http://localhost:631 typed into your favorite browser. The interface is shown in Figure 8.1. In addition, there is a CUPS application program available which does not use a browser, but the look and feel is identical.

There are some pages on the GUI that will require a username and password to perform some actions on them, such as to add a printer. On most Linux distributions, only root can add, modify, or delete a printer or class of printers.

Exam Warning

You must remember the Port names and names that are used commonly used for the exam. The Web interface port is one of the required ports to remember.

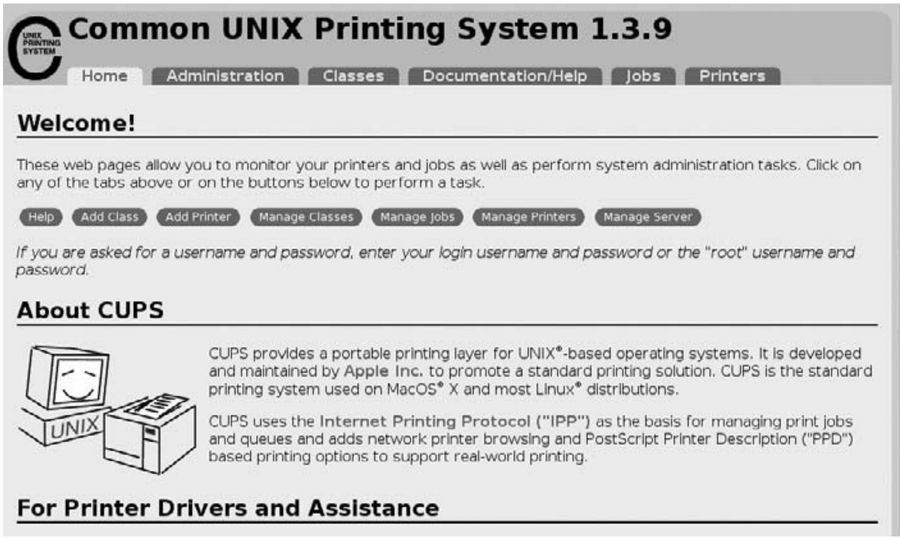


FIGURE 8.1
CUPS Interface in a browser.

Managing Printers, Jobs, and Queues

The CUPS management interface can be used to add and delete printers, and this is accessed under the **Administration** tab of the GUI. This may require root privileges to run. Local printers can be added directly if you know the make and model, or the system can find new printers that have been attached to it.

Printer jobs can be managed through two options: directly from the first page in the **Administration** tab, or by clicking on **Manage Printers** under the **Administration** tab and managing the jobs from there. The management of printers section is very powerful and allows the user to

- print a test page.
- stop/start a printer.
- enable and disable a print queue.
- move jobs from one print queue to another.
- cancel all jobs.
- set printer options.
- allow specified users.
- set as default printer.

Later on in this chapter (in the "Printing Commands" section), we will look at the command-line versions of some of these commands. The CUPS interface is very easy to understand, and for the majority of users it is easier to use than the command line. As the system administrator can prevent unauthorized users changing certain functions, it provides an easy, safe GUI to deploy to users.

Learn by Example: Adding Banners to All-Print Job

One printer in your work area is used to print confidential documents, and you want to make sure that a banner page is printed between each job to remind users of this fact. You access the CUP interface on the remote server using http://10.10.100.134:631. Click on the **Administration** tab and browse to the printer you want to change. Click on **Set Printer Options**, entering the superuser name and password when prompted. On the next page, select **Confidential** as the starting or ending banner, and then click on **Set Printer Options**. You will now get banners between print jobs.

CUPS Printer Classes

CUPS allows the user to group printers together, and this collection of printers is called a *class*. This is particularly of interest in a company which has a number of printers and they can be grouped together in areas, such as finance, admin, and so forth. This allows printers to be taken offline for maintenance, and the user will not notice any disruption or need to remember the names of different printers in their area. The adding and management of printer classes is undertaken in the **Administration** tab of the CUPS GUI.

EXERCISE 8.1: Testing a Printer Through CUPS

In this exercise, we will test a printer to ensure that you can print to it.

1. Initially, go to the printer, and confirm it is switched on and has paper in it.
2. On your system, open up the CUPS GUI in a browser using the URL http://localhost:631, and click on the **Administration** tab.
3. Click on **Manage Printers** and scroll down (if necessary) to the printer you are trying to test.
4. Check the printer state to see if it is idle and accepting jobs. If it is, click on **Print Test Page** and see if the printer prints it correctly.
5. If it does, your printer is set up correctly.
6. If no page is printed, then check if you are sending the page to the correct printer. ■

Printing Commands

Once a printer has been installed and properly configured, a user is able to print to it from any printer-capable graphical client. Text-based interface can also print through a simple command-line interface. These basic commands are described below.

lpr

The `lpr` command submits print jobs to the specified printer, or the default printer if none is specified. The main options that a user will need are shown in Table 8.1.

Table 8.1 `lpr` Printing Options

`-P destination`	The name of the printer to send the job to.
`-# number`	Print a specific number of copies (default is 1 copy).
`-T title`	Prints a title on the banner page of the output.
`-h`	Suppresses printing of the banner page.
`-w cols`	Prints file with pages of a specific width.
`-m`	Send mail when the job has printed.
`Filename`	The name of the file which you want to print.

This command is used with BSD UNIX systems, such as PCBSD, NetBSD or FreeBSD.

lp

The `lp` commandis very similar to the `lpr` command above, but works on System V systems. There are a number of differences in the syntax, such as the `-o` parameter with `lp` has a number of different options such as `nobanner`, `cpi=pitch`, and `width=chars`. Depending on which system you are working on, you should understand the syntax of each command.

> **Note**
> With CUPS installed on your system, both of these command line interfaces are supplied. This means that a user on the system can use either `lpr` or `lp` to print. The advantages of this are that users will not get confused moving from one system to another, and scripts that have been written with either of these commands will work correctly.

lpq

When a user wants to see the status of one or more print queues, the `lpq` is used. This command can be run once or displayed continuously at a specified interval until the queue is empty. The command on its own will display the queue of the default printer. The main options for the command are as follows:

`-a` to show the queue status of all printers

`-P` to show the status of a specific printer

`+interval` will display the queue every *interval* seconds until empty

Table 8.2 `lpstat` Options

`-a`	Displays the queues for all printers.
`-d`	Displays the default destination.
`-h server`	Specifies the CUPS server to communicate to.
`-o option`	Displays the queue on printer.
`-p printer`	Shows status of printer.
`-r`	Status of CUPS server.

lpstat

The `lpstat` command can show the status of printers and queues like the `lpq` command; however, there are far more options, including the options to query the status of specific CUPS servers in the network. The options are shown in Table 8.2. This is useful for system administrators who can administer print servers and queues from a central location.

cancel

Users or administrators can cancel jobs still in the queue of a specific printer. An individual job can be canceled, or all jobs can be removed (if the user has the appropriate rights). The default printer queue will be checked if no specific printer is defined.

X11

X11 is the common name for the X Windows system, which is the GUI found on most Linux distributions today. The origins of the X Windows system date back to 1984, when the protocol was developed by MIT, and the current protocol version (X11) appeared in 1987. The current implementation of the standard is overseen by the X.org foundation, which came into being in 2004. Apart from Linux distributions, it can also be found in Cygwin/X running on Microsoft Windows, Sun Microsystems' Solaris, and with the latest version of Mac OS X.

Starting and Stopping X11

There are a number of ways to start and stop the X Windows system. The display manager will normally be loaded automatically on boot. This is accomplished by setting up the /etc/inittab file to load the X windows system, when multiuser mode is used (run level 5). The actual display manager that

will be used is defined in the /etc/sysinit/displaymanager configuration file, as shown below for the KDE environment:

```
DISPLAYMANAGER="kdm4"
```

When the Linux system is booted into a multiuser mode without a graphical login using the display manager, such as runlevel 3 in the /etc/inittab file, you will have to start an X session from the command line. The X server and X session must be started, and this can be accomplished using `startx`, often without any parameters.

The `startx` command will use configuration settings found in the .xinitrc file, which by default will be in /etc/X11/xinit directory; however, the user can customize the X session and launch default clients using a .xinitrc file located in their home directory. If you intend to use the display manager that was installed with your system, you do not need to use a customized .xinitrc file.

The `startx` command script can be used to pass parameters such as for color depth, whether the display adapter is a multiheaded device, or the resolution (expressed in dots per inch (DPI)) of the monitor.

The `startx` command itself can be passed with parameters that it passes to the X server before an X session is launched. One such parameter is the `-depth` option, which will alter the number of colors used. This can be used during development or testing of X clients to ascertain how each will look on different hardware solutions. This may be useful if you are planning a large roll out of Linux across a variety of differing hardware. This will start an X session a depth of 16.7 million colors.

```
$ startx -- -depth 24
```

The `startx` command can also be used to launch multiple X sessions in a number of virtual consoles. The complexity of multiple X sessions means that it is not an everyday occurrence, and you may wish to be very familiar with X sessions before you try this. Each X session then can be closed separately or the system restarted in run level 3 as above.

Difference between X11 Clients and Server

The X Window System was developed using a client-server model, which means that the two components can be distributed on separate systems. On a typical Linux configuration, both these components are located on the same machine; however, they communicate through the standard processes. In a client-server application, the client software process will initiate a communication session, while the server waits for a request from one of its clients. Most people use client-server programs on a regular basis without realizing it. For instance, a Web browser is a client program that requests data from a Web server.

In X11, the X server communicates with various client programs. This server accepts requests for graphical output (that is, the normal display window on a screen) and also accepts user input (from, typically, the keyboard and mouse) and sends these back to the client. The server can send out to a display on another system, or may control the video output on the local system. Thus, the user's terminal is the server and the applications are the client, which is often the cause of confusion for new users as they typically think of a server, from their perspective, as an end-user. Their normal perception is that the client runs on the users' computer and the server is remote. In X Window terminology, the server provides display and I/O services to the applications. The applications that use these services are clients (such as a browser).

The server and client communication protocols run with network transparency; that is, it is invisible to the applications that are using it. In a workstation configuration, these will typically be on the same physical machine. When Linux is designated as a server, the clients are often distributed across the network. This communication protocol can be tunneled over an encrypted tunnel. X can be bandwidth-intensive, so a fast network is desired, particularly if there a lot of remote clients.

Window Managers

The X Window manager is the windowing system that runs on X, and the user can choose one of many window managers. These Windows managers will have certain requests between the client and server redirected through them, such as when a new window needs to be displayed. The core X Windows System does not include icons, so these are specifically maps by the window manager. The types of window manager are large, and include *tiling* window managers (such as *ion*, *dwm*, and so forth); *composting* window managers (GNOME and KDE are good common examples); and *stacking* window managers, such as *IceWM*. There are also *virtual* window managers, which use virtual screens whose overall resolution can be greater than the monitor they are displayed on. No one window manager is better than another, as each has its own purpose, and which one you use will comes down to personal taste. For the majority of more popular distributions (Fedora, OpenSUSE, Ubuntu, and so forth) that are used by the general user, either GNOME or KDE will be installed, often with the selection available at installation time.

In general terms, the window manager controls the appearance of the GUI to the user. It positions the windows, controls the fonts and colors, and handles the input and output (mouse clicks, and so forth). The window manager is just another client from the point of view of the X window server. The initial window that is displayed is defined as the root window, with top-level windows being children of this root window.

X Session Manager

The state of the desktop or session at a given time is managed by the X Session Manager. This allows, for instance, a user to log out and then log out of a session, and to have the same windows displayed. The session manager stores the state of all the windows on exit to restore them. While the default session manager *xsm* can be used, specific session managers are often bundled with the display manager, such as *ksmserver* in KDE.

Display Manager

The X display manager runs as a program that allows the X server to start a session on a system, either local to the server or remotely. The display manager often prompts the user for a username and password with an initial login screen, although it may be bypassed depending on the setup of the system (usually through KDM's auto login feature). When it is run on a local system, it will start the X server before presenting the login screen. In a remote session, the display manager will act like a Telnet server, requesting a username and password and starting the remote session. The default display manager in X is the X Windows Display Manager(XDM). The most popular alternative display managers are explained below. Alternative display managers are often used as the default XDM is often more complex to configure for the standard user.

KDE Display Manager

The display manager bundled with KDE is the KDE display manager (KDM). The underlying toolkit that KDM is based on is, like KDE, the open source software toolkit Qt. It is very configurable from the KDE control center, allowing screen color, menu options, styles, and so forth to be configurable. KDM was originally based on XDM.

The main configuration file for KDM is often found in */usr/share/kde4/config/kdm*, and is called *kdmrc*. KDM must be run before a user is logged in; so it is, therefore, not associated with any user, and hence user-specific configuration files are not possible. Users can, however, change the appearance of their desktop once they are logged in.

The graphical login manager within KDM is *the greeting*, which can show a company logo, current system time, or perhaps nothing. Some or all users may be allowed to shut down the system from this initial screen.

GNOME Display Manager

GNOME was conceived slightly later than KDE, and again uses open source software. The GNOME display manager (GDM) uses *metacity* as a default. Unlike KM, this was written entirely from scratch and does not contain any

original XDM code. Upon startup, the GDM daemon reads its local configuration file, gdm.conf. When there is more than one local display, each of these forks an Xserver and slave process. The initial process will start the display, and calls `gdmlogin` to prompt the user for a username and password. Remote displays are managed using the X Display Manager Protocol (XDMCP), typically running on port 177.

The configuration file can be found in the file /etc/gdm/gdm.conf, and the syntax within the file follows the standard GNOME file syntax. The file contains a large number of options for the daemon configuration, security, and remote and local GUI. When the system is installed, most of these are set up automatically. The look and feel of GDM can be configured by a large number of themes, which are available in the main package or downloadable from various sites.

Differences between KDM and GDM

The proponents of GNOME, KDE, or any other similar display managers can be found everywhere. At first glance, the differences between GDM and KDM are purely cosmetic (colors, whether there is a toolbar at the bottom of the screen or main menu, and so forth). Ignoring the base colors (which can easily be changed on either system), we will discuss the differences between the two managers. It is often said that, overall, a system with KDE installed is more Microsoft Windows-like than the one with GNOME.

GDM has, by default, two toolbars at the top and bottom, and also splits it menu into three submenus: **Applications**, **Places**, and **System**. KDM will have only one toolbar at the bottom, and its menu will be slit into **Favorites**, **Applications**, and **Computer**. In terms of user input, the default for KDM is one click, and two in GDM. Also, the system configuration menus are usually more complicated in KDM as opposed to GDM.

You can install both window managers onto a system and switch between them (or have them display on different monitors at the same time if you have enough memory and computing power). This is undertaken in many ways, with some common distributions like Fedora using the `switchdesk` command.

Multiple Desktops

Within the X Windows systems, there is the ability to have more than one desktop. Each desktop is known as a *virtual desktop* within the system, and each can operate independently of one another. This concept allows the user to create two or more separate environments, where simultaneous tasks can be undertaken. This may involve a word processor in one desktop, your

e-mail client open in another, and perhaps some scripts running in a terminal session in another. The use of virtual desktops allows an uncluttered view of the application your need to concentrate on and change to another desktop at the click of a button.

KDE Virtual Desktop

There are four virtual desktops installed with KDE as a default, with the option to install a total of 36. These are shown on the bottom toolbar, numbers 1 to 4, as shown in Figure 8.2.

Moving between desktops is achieved just by clicking on the appropriate number. The standard desktop is initially installed on all desktops. To make it easier for navigation, these can be renamed using the multiple-desktop control module shown in Figure 8.3, with the new toolbar displayed in Figure 8.4.

FIGURE 8.2 *Default KDE toolbar showing virtual desktops.*

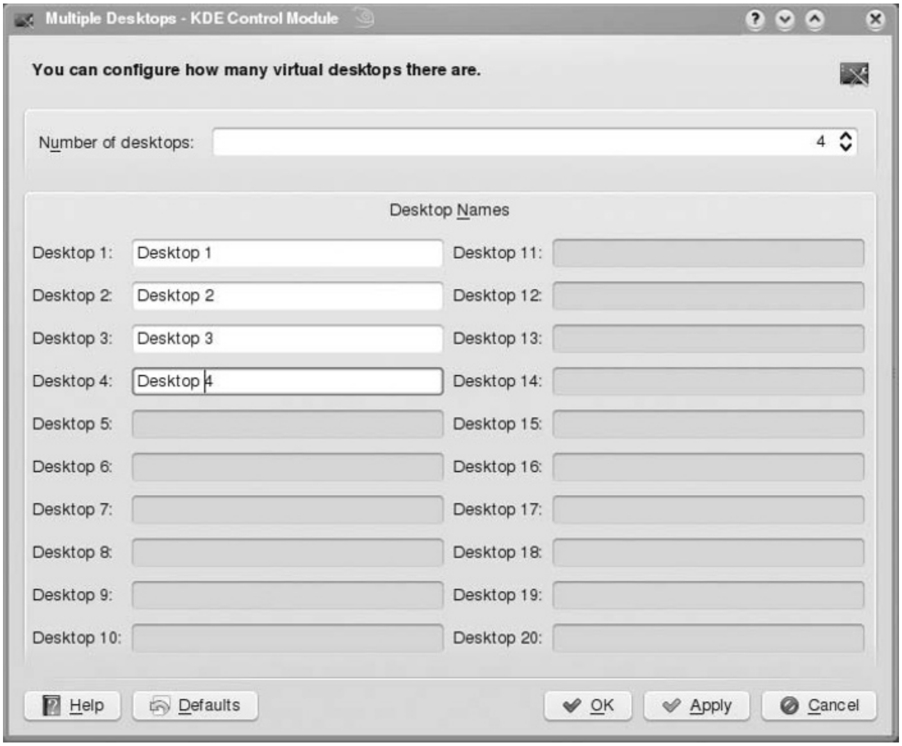


FIGURE 8.3 *Multiple-desktop control module.*

FIGURE 8.4
KDE toolbar showing named virtual desktops.

FIGURE 8.5
Default GNOME workspace.

The toolbar will then appear as shown in Figure 8.4.

GNOME Workspaces

Within the GNOME system, these are called *workspaces*, and there is a *workspace switcher application* which controls this. You can set different preferences for this, including the number of workspaces you want (up to 36). The user can move to a different workspace using the applet, or you can move to one by pressing **Ctrl** and scrolling the mouse to highlight the desired workspace. A part of a typical GNOME screen showing four numbered workspaces is shown in Figure 8.5. Rather than having names such as Desk 1, Desk 2, and so on, you can rename the workplaces to more meaningful names – perhaps mail, scripts, Internet, and so forth. Individual applications within one desktop can be moved to another desktop or can be made visible in all workspaces.

EXERCISE 8.2: Configuring Your Desktop

1. With both GNOME and KDE, you are given four desktops by default, and you should start to use these to realize their potential. When you first start Linux, think about what will be your main task – say, Web browsing, preparing documents, and e-mail.
2. Click on the first desktop, open up your favorite browser.
3. In the second, open up all the word processing tools you need; and in the third, start your e-mail client.
4. Then, open up the configuration tool for your environment (for example, multiple desktop manager in KDE) and rename the first three virtual desktops to match what you have started in them.
5. This will make your desktop less cluttered and easy to navigate. ■

X Window System Directories

The current implementation of X, as stated before, is now controlled by the X.org foundation. This distribution is commonly called Xorg, and is a sizable piece of software – typically over 100 MB, and possibly is double this size.

> **Note**
> The X Window system can contain additional clients, an array of icons, as well as many games. As the majority will be located in a subdirectory of the /usr directory, this may be a problem if you are installing this in an older machine. The disk space on modern machines is not usually an issue, but care should be taken if you size the partitions yourself.

The /usr directory and its subdirectories contain the majority of Xorg's software, with the most significant directories being:

- **/usr/bin** – This contains the X server and the clients that are installed.
- **/usr/lib** – This is where the software libraries are held in support of the X server.
- **/usr/include** – This has all the included files needed if a new X client needs to be developed.
- **/usr/lib/x11** – A symbolic link will be found to this directory in /usr/lib. It contains the font file, documentation, system, and client resources.
- **/usr/lib/modules** – This will hold all the drivers for various graphics cards used by the X server.
- **/usr/X11/man** – This is, as you would expect, the man pages for X11.

When Linux is installed in a computer and the graphical desktop option installed, the main components required to run a local X session are loaded. These basic components are the X server, basic fonts, a terminal client, and a window manager (described in more detail in earlier sections). The main configuration file for the X Windows system is the *xorg.conf* file. The xorg.conf file is typically located in /etc/X11/xorg.conf, although this does vary across some Linux distros. In early versions of X, this file had to be edited and manipulated manually, especially for unusual input devices and when multiple monitors were in use. This was far from ideal, and modern systems with a number of new extensions integrated in the X server.

The xorg.conf file will contain a number of different sections, which will provide information on the input devices, monitor layout, file locations, and so forth. The main sections are as follows:

- **Files** The location of the fonts and colors of the X server.
- **Module** Informs the X server on the dynamic modules to load.
- **InputDevice** It defines all the input devices, such as the mouse and keyboard that are used in the system.
- **ServerLayout** It defines the display and defines the screen layouts.
- **Monitor** The attributes of the monitor(s) that are attached to the system.
- **Screen** The attributes of the screen configuration.
- **Device** It defines all the graphics cards attached to the system.

The *ServerLayout* sections are at the highest level, and bind together the input and output devices to be used in a session. The output devices will comprise of a number of sections (for example, the graphics board and a monitor). All the input and output devices are bound together in the *Screen* section, and these are used in the *ServerLayout* section. While most of the settings for the input and output sections are configured automatically, some users may wish to override these (for example, to force the monitor to a lower resolution).

Exam Warning

The name and location of the main configuration files are important to know for the exam. In particular, make sure you know the configuration files you need to change to configure individual user's desktop.

Terminal Emulators

Within Linux, there are a number of built-in terminal emulators, often bundled with the XDM. The two common ones are `gnome-terminal` and `konsole` in GNOME and KDE, respectively. These are feature-rich and have a very user-friendly interface, although both have a relatively high memory footprint. For everyday tasks, where memory is not an issue, these terminal emulators are very good and easy to use, and are probably the preferred

interface for a user. However, for systems that have a smaller amount of random access memory (RAM), there is another emulator called `xterm`, which has a very small memory footprint (usually under 1 MB), although the window is smaller and the default font is hard to read. The standard xterm emulations are for DEC VTxxx and Tektronix 4014 terminals.

The VTxxx and Tektronix terminals can each have their own window so that a user can edit text in one and look at graphics in another. Only one of these windows will be considered active at any one time, and this one will accept keyboard input and terminal output. The number of options allowed for xterm is very large, which will allow the system to correctly emulate the terminal and application running in it.

SUMMARY OF EXAM OBJECTIVES

In this chapter, you have learned about how to configure a Linux system when it is used as a workstation. The two most common aspects of the user experience – namely, printing and the GUI (X Windows interface) – were explained, and how these can be configured for individual users.

In spite of the move toward the paperless office, printing letters and other documents is still a firm requirement for most users. With the move to digital photography, the home user is now often using their printer more than in previous years. The setup and use of a printer in Linux is slightly more difficult than in the "plug and play" world of Microsoft Windows, but the use of the CUPS interface is now making this a much easier task.

The CUPS interface was shown, and how to add local printers using the interface. The management of printers and their queues was demonstrated using the CUPS interface, and also how this could be achieved using the command line interface.

The users' main interface to the computer is through the GUI, and the number of different interfaces that can be used is very large. The standard, underlying interface is X Windows, currently at version 11, and is commonly known as X11. This can be used and enhanced with many different display managers, the two most common ones being GNOME and KDE. The main configuration files for X Windows were described, and how to modify these for individual users. Specific configuration details were then outlined for the GNOME and KDE desktops. The use of multiple or virtual desktops were explained, and how these may be different from a user's experience of a single desktop. The configuration of these multiple desktops was explained.

SELF TEST

1. A user has sent three jobs to a printer, with job numbers 372, 373, and 374. They now want to remove printer job number 373, which has not been printed yet. Which command will achieve this?

   A. `lpr --cancel 373`
   B. `lpstat -c 373`
   C. `cancel 373`
   D. `lpr -c 373`

2. A user is running KDE as his display environment. What will be the most likely environment variable for the display manager?

   A. DISPLAYMANAGER="KDE"
   B. DISPLAYMANAGER="kdm4"
   C. DISPLAYMANAGER="kde_display"
   D. DISPLAYMANAGER="gdm"

3. A user is sending a job to printer EPSON_COLOR, which is not his or her default printer. They want it printed with a banner title of *myjob* and then a mail to be sent to let them know when it has been printed. The user will need to use the following command:

   A. `lpr -m -T myjob -P EPSON_COLOR`
   B. `lpr -sendmail -C myjob -P EPSON_COLOR`
   C. `lpr -m -T myjob`
   D. `lpr --sendmail -T myjob -P EPSON_COLOR`

4. A normal user wants to disable a printer in the CUPS Web interface. What do they need to do to achieve this?

   A. Enter the superuser username and password when prompted by CUPS.
   B. Open a terminal window and enter the superuser name and password before launching the Web browser.
   C. Start CUPS with the `-s` option.
   D. Start CUPS, and enter the superuser name and password in the **Authentication** tab.

5. The main X Windows configuration file xinitrc is likely to be located in which system directory, when GNOME has been installed as the only display manager?

   A. */etc/X11*
   B. */X1*

C. */usr/X11*

D. *User's home directory*

6. Which of the following would be the best description of the X Windows System if you were describing it to a new Linux user?

A. X Windows is a client-server architecture, with the client accepting keyboard input.

B. X Windows is a client-server architecture, with the server accepting keyboard input.

C. X Windows is a client-server architecture and cannot be ported to run on a Microsoft Windows system.

D. Both the X Windows server and client must be on the same system.

7. A system administrator wants to add remote displays to systems configured with X Windows. Which protocol and port will be used between the X server and the remote client?

A. *XDMP* normally running on port 177

B. *XDMCP* normally running on port 177

C. *XDMP* normally running on port 187

D. *XDMCP* normally running on port 187

8. A user has installed Linux on his or her system and has made KDE as the default desktop manager. What is the default number of virtual desktops, and the maximum number that can be configured by the user?

A. Default of four desktops and a maximum of 36

B. Default of two desktops and a maximum of 36

C. Default of two desktops and no maximum

D. Default of two desktops and the user cannot configure any more

9. A company has installed Linux with the GNOME desktop on a number of older systems as a means to extend their life. These systems use an 800×600 display and have a maximum of 64 MB of RAM installed. They want to use these systems as remote terminal emulators to a more powerful X server on another system. Which remote terminal would give the best performance due to its small memory footprint?

A. `kconsole`

B. `gconsole`

C. `gnome-terminal`

D. `xterm`

10. A user is configuring his or her Linux system, which has KDE installed on it. The user wishes to add a new printer and will do so through the CUPS Web interface. The user has installed Firefox, and the CUPS server on the system and both are working correctly. What would be the best way to access the CUPS server?
    - **A.** CUPS can only be accessed from the command line using the command `CUPS -S` when it is installed locally
    - **B.** CUPS can be accessed using Firefox with the URL http://localhost:631
    - **C.** CUPS can be accessed using Firefox with the URL http://631:localhost
    - **D.** CUPS can be accessed using Firefox with the URL http://cups@localhost

11. A user has problems with the startup of his or her system, and the user wishes to start up the system in single-user mode and then to start X Windows. Which is the best method to achieve this?
    - **A.** Start the system in run level 1, and then run `startx`.
    - **B.** Start the system in run level 5, and then run `startx`.
    - **C.** Reboot the system, and when the initial load screen appears, type **Ctrl** and **S** together.
    - **D.** Reboot the system, and when the initial load screen appears, type **Ctrl** and **1** together.

12. You have just downloaded and installed the latest version of the GNOME desktop. This is a beta version, and your system seems to freeze when you start it. Which option would be worst one to use?
    - **A.** Press the **reset** button, and boot into single-user mode with command-line input. Then, remove the beta version.
    - **B.** Open a terminal window and type `shutdown -now`, and then boot into single-user mode with command-line input. Then, remove the beta version.
    - **C.** Type **Ctrl** + **Alt** + **F2**, and use the root username and password when prompted. Look at the PID list and kill all the processes associated with X Terminal session. You can now uninstall the beta version.
    - **D.** Type **Ctrl** + **Alt** + **F7**, and use the root username and password when prompted. Type `rollback X11` to revert to the previous version of X Windows.

13. You want to run the GNOME window manager if you boot your system into runlevel 5, and the Openbox window manager if you start the system in run level 4. How can this be best achieved?

A. You cannot start a different window manager based on runlevel.

B. Modify the .xinitrc in your home directory, and include shell code to execute the commands `exec gnome-session` or `exec openbox-session` (based on the current runlevel).

C. Boot the system directly into a terminal session and run a script to start X (based on the appropriate runlevel).

D. Add the lines in the .xinitrc file in your home directory to switch between window managers:

```
if runlevel=4 then
 gnome-session else
 openbox-session
 end
```

14. You are a system administrator for a large company, and a user wants to purchase a new color printer for his administrative assistant to produce sales literature. This printer is not the usual printer you purchase. What would be your best advice to the user to ensure the new printer works with his or her Linux system?

A. Find out the make and models of the printers he is considering purchasing, and check the www.cups.org Web site to see if the printer's drivers can be downloaded.

B. Look at the printer manufacturer's Web site to see if the printers are listed as "plug and play" devices, and hence will work seamlessly.

C. Tell the user he can buy any printer that has the "CUPS Compatible" logo on the box.

D. Tell the user he can buy any HP printer, as they are all compatible with Linux through downloads on HP's Web site.

15. You have sent a job to your default printer and have seen that there are a lot of jobs before it. As you need the printout in a hurry, which is the best option?

A. Using the CUPS GUI, find an idle printer and move your job to this printer.

B. Move your job to the top of the queue on your default printer using the CUPS GUI.

C. Login as superuser on the CUPS GUI. Pause all the jobs on the printer ahead of your job so your job will start next.

D. Resend your job to the printer using the option `-priority` on the `lpr` command, which will insert the job to the head of the queue.

SELF TEST QUICK ANSWER KEY

1. C
2. B
3. A
4. A
5. D
6. B
7. B
8. A
9. D
10. B
11. A
12. C
13. B
14. A
15. A

Installing, Configuring as a Server

Exam objectives in this chapter

- Network Services
- Web Services
- Application Services

UNIQUE TERMS AND DEFINITIONS

- **Apache** It is an open-source Hypertext Transfer Protocol (HTTP) (Web) server produced by the Apache Software Foundation that has become the most widely used Web server on the Internet. It aims to be aligned with current HTTP standards and to run on all modern operating systems.
- **Web proxy** It is a server that acts as a go-between between a client and typically the Internet, often to perform filtering of the data. Squid is a good example of a Web proxy server.
- **Domain name system (DNS)** It is a hierarchical naming system for computers, services, or any resource connected to the Internet. It associates information, such as IP addresses, aliases, and resource types, with domain names assigned to each device or service. Its most important task is to resolve IP addresses with domain and host names and vice versa.

- **Network Time Protocol** NTP is the Transmission Control Protocol/Internet Protocol (TCP/IP) used to synchronize the clocks on computers across a network. NTP uses User Datagram Protocol (UDP) on port 123.
- **MySQL** It is an open-source relational database management system that is developed, distributed, and supported by Sun Microsystems, Inc.

INTRODUCTION

The role of a server is to act as a central repository for data and to serve applications and services to local and remote clients. The server can be specific to one particular need, such as a Web server or print server; but often in small networks, the servers are multifunctional. The server can serve network services to clients, such as DNS and Dynamic Host Configuration Protocol (DHCP), or as an application server for commonly used services such as printing and mail. This chapter will explain how to configure these services.

Companies need to have a Web presence, and the setting up of the Apache Web server along with some of the additional modules that need to be added to it (PHP, Common Gateway Interface [CGI], and so forth) is defined. Testing of the Web server using the command-line interface is outlined. While it is not considered very secure into today's world, the setting up and use of a File Transfer Protocol (FTP) server is defined. Some methods to make this server more secure are outlined. Finally, the Squid proxy server is described, and how it can be used in a network to help speed up the throughput is also described.

Finally, the main applications that you may wish to run on a server are explained – printing, mail, and a database. The popular variants of these are outlined, along with details on some of their configuration options. The use of mail servers and how to secure these from spammers is discussed, along with the different methods that clients can connect to them.

NETWORK SERVICES

The basics of setting up servers to act as a DNS and DHCP server are explained, along with an introduction to their configuration files. Most companies would benefit from installing a DHCP server into their environment to manage the allocation of IP addresses, among other data. The basics of configuring and testing a DHCP server are explained, and a sample DHCP

server configuration file is shown as a guideline. The need for a Web presence is essential in today's business world, which will require the setting up of DNS entries, either locally or utilizing a third party. This chapter will describe how to set up a local DNS and explains how a basic configuration would look like.

The synchronizing of time throughout a system is very important to ensure files are created correctly across servers. An incorrect time on a system may result in a system not being able to synchronize files between two servers correctly, potentially allowing a file that is marked as older being overwritten when it was in fact the most recent. In addition, the examination of log files during an investigation can be more difficult when there is no consistent time across a network. The setup and use of an NTP server and the methods that are used to keep that time server in sync are explained.

Interoperability with Microsoft Windows is explained, and how to set up and run a Samba server to achieve is explained. The coexistence with Microsoft Windows is very important, as most business networks will have a number of these clients and servers installed on them. The number of all Linux networks is small, and therefore this interaction will occur on a regular basis. Companies will often have Microsoft Windows clients interacting with Linux and Microsoft Windows servers, and it is therefore necessary to present to the end user a seamless interface to both servers.

When servers are primarily providing network services, these are often located locally but in restricted (and often cold) server rooms. Administrators may want or need to connect to these systems remotely. A number of common remote access utilities, such as virtual network computing (VNC) and remote desktop, are explained, and how these can best be used are also explained.

Dynamic Host Configuration Protocol

A DHCP server is used to configure hosts to work correctly on your network. A system administrator can configure the TCP/IP parameters for a host as they connect to a network (of course, after the NIC is activated). These parameters will include the following:

- IP address
- Setting one or more name servers in /etc/resolv.conf
- Configuring the routing, including the default route

A DHCP server will allocate an IP address on a permanent or temporary basis. IP addresses, which are allocated on a temporary basis, are said to be

leased; and at the end of the lease, the network client can extend or relinquish the lease. The advantage of DHCP for a company is that the method of assigning IP address is automatic; hence, a duplicate IP address is unlikely to be assigned. In addition, if you have limited IP addresses, DHCP can maximize the use of each of them by allocating addresses, as and when needed. Servers are not usually configured using temporary DHCP leases, as you will always want these to be *static addresses* so that they can be linked with DNS. The DHCP server maintains a list of the leases in a file called *dhcpd.leases*, usually in /var/db.

When you install Linux as a server, the option to install the DHCP server software will be listed. After installation, the DHCP server can be added easily from a number of Web sites or via the *install software*, which will be present in most distributions. The complexity in setting up of a DHCP server will depend upon your network architecture and the amount you want the DHCP server to do. The following will outline the basics for setting up the server, which will be a good basic introduction into the subject.

DHCP Server Configuration

Before starting to configure the DHCP server, it is very important to understand your network and what parameters you will be configuring. The following are the basic parameters you will need to configure:

- Domain name
- DNS servers
- Lease times
- Routing
- Static IP addresses
- Logging
- Primary or secondary DHCP server

If you have a large network with many servers, you may wish to allocate a contiguous subnet for these servers to make the allocation of IP addresses easier. In addition, there will be some users who need a static IP address, perhaps to allow them to have specific access rules in a router or firewall. These should also be grouped together whenever possible to make the configuration tables more readable. The configuration file for DHCP is /etc/dhcpd.conf, and a sample file is included when you install the DHCP server.

Each subnet that you are going to provide DHCP services for must be defined in the file. The main options that can be used are described below,

although there are many more, and the complete list can be found by typing `man dhcp-options`.

```
#
#sample configuration file for dhcpd
#

#option definitions common to all supported networks...
option domain-name "example.org";
option domain-name-servers ns1.example.org, ns2.example.org;

#set the time a client can keep the IP address
default-lease-time 600;
max-lease-time 7200;

#if this DHCP server is the official DHCP server for the local
# network, the authoritative directive should be uncommented.
#authoritative;

#set the default gateway to be used by clients
option routers 10.254.239.1;

#set up the NTP server
option ntp-server 10.254.239.6;

#set the nameserver to be used by the clients
option domain-name-servers 10.254.239.5

#set up a WINS client for Microsoft Windows clients
option netbios-name-servers 10.254.239.3;

#this is a very basic subnet declaration.

subnet 10.254.239.0 netmask 255.255.255.224 {
 range 10.254.239.10 10.254.239.20;
 option routers rtr-239-0-1.example.org, rtr-239-0-2.example.org;
}

#a network set-up for future use, but not currently used.
subnet 10.254.240 netmask 255.255.255.0 {
}

#fixed IP addresses can also be specified for hosts.
#names or IP addresses can be used
```

```
host adminprinter{
 hardware ethernet 08:00:07:26:c0:a5;
 fixed-address adminprinter.fugue.com;
}
```

As can be seen, this setup is allocating a number of IP addresses using DHCP, and there are a number of IP addresses that are not in scope. Even if you own a large subnet, it is often worth using a portion of these if that is all that is required. This will make it easier in the future to allocate static addresses for additional servers or to subnet part of the range off for other uses. The file also shows the setting up of a fixed IP address; in this case, showing how to allocate a fixed IP address to a printer called *adminprinter.* The hardware Ethernet address of the printer needs to be known and set up in this file.

The above will enable you to start the configuration of your DHCP server. The full range of options is far beyond the scope of this section, and users who wish to know more are encouraged to look at a number of books on the subject. One useful text can be found at www.dhcp-handbook.com.

Domain Name Server

The DNS resolves machine names to e-IP addresses, either in IPv4 or in the newer IPv6 standard, or it converts from the IP address to the name. In principle, a DNS resolves a name such as www.syngress.com to its IP address (in this case, 145.36.40.200). The DNS specification is defined in a number of standards, but especially RFC1034 (Domain Names – Concepts and Facilities). Request for comments (RFC) is the method of generating standards for and about the Internet. For those who wish to know about DNS than that is covered here can look at any of the many books on the subject.

The Domain namespace can be regarded as structured in a tree form. Each domain within the tree is a node, with each node having a set of resource records associated with it. These records will define the ownership, name, and IP address, as well as potentially a lot of other details. In addition, each of these domains can have subdomains, often referred to as children, associated with them. The root of this tree is named "." (dot), which is the corollary of the root drive/in the Linux filesystem.

Each subdomain prepends its name to the root name, each being separated with another "." or dot. The root domain is therefore ".", with *com.* and *org.* being examples of subdomains. There are also international subdomains such as *.uk* for companies in the United Kingdom. Each of these will have further subdomains, such as *syngress.com*. These could have other child

domains as well, such as Elsevier is a division of Syngress, and as such could be defined as *elsevier.syngress.com*. Actual machines in each domain can be defined by their machine name and the domain they are in, which will give the *Fully Qualified Domain Name* (*FQDN*). This specifies the exact position in the DNS tree structure. In DNS, this will be specified with the trailing dot, such as *webserver.syngress.com.* will define a machine whose name is *webserver* in the *syngress.com* domain. This will be a unique definition; although there may be other machines with a name of *webserver*, there can only be one *webserver.syngress.com.* in existence.

The Domain name system is a distributed database, which uses a client-server model. The nodes of this database are *nameservers*, and each domain and subdomain has one or more authoritative DNSes that publish all the information about the domain. All this information is included in *DNS zone files* on the DNS server or *nameserver*. There can be forward and *reverse* zones defined. A reverse zone is used to associate an IP address with a host-name. A forward zone is used to associate a name with an IP address. The DNS resolver is the client part of the client-server architecture and is the process that performs the resolution of the query; for example, the translation of the FQDN to its actual IP address.

DNS Resource Records

As stated in the section above, the resource records (RR) are the most basic part of the Domain name system. These records have a number of basic elements to them: type, time to live (TTL), class, and possibly some data specific to the type of record. These records are described in Chapter 5 in the section "DNS Record Type and DNS Resolution."

Caching Nameserver

If you install and configure a caching nameserver, this will build a local cache of resolved domain names and will use this list to serve other hosts on your network. In practice, a large number of DNS requests are the same, and this will increase the speed of resolution and decrease the amount of traffic you send upward to another nameserver. The `named` server program available on Linux systems, which is part of the BIND package, will be able to provide these services. Prior to version 8, BIND uses a configuration file based on /etc/named.boot, while later versions use /etc/named.conf. We will look at the earlier version because of its simplicity and the fact it is used more often.

The /etc/named.boot file is usually a small file with pointers to master files containing zone information and other information. One of the first lines in the file will be a statement specifying where the main files are located:

```
directory /var/named
```

There will then be a series of lines to specify the primary and secondary servers in the domain, along with a cache statement.

```
;
;         domain                      file
cache     .                           named.ca
primary   mydomain.com                named.hosts
primary   0.0.127.in-addr.arpa        named.local
primary   10.100.100.in-addr.arpa     named.rev
secondary my2dom.com                  192.168.10.6 named.hosts
```

The version of BIND from 8 onward will have the same information but in a slightly different format. A sample listing showing how a primary or master nameserver is set up in the later version is shown below, which is in the *named.conf* file (normally in /etc or /etc/named.d):

```
zone "mydomain.com" IN {
   type master;
   file "mydomain.com.xone";
   allow-update [none;];
};
```

With the later version, there is a control utility called `rndc`, which allows you to administer the `named` daemon. The configuration file for `rndc` is /etc/rndc.conf, and additionally, you need to specify authentication keys in both /etc/rndc.conf and /etc/named.conf, which must match. You will need to generate HMAC-MD5 keys for both configuration files using the following command:

```
dnssec-keygen -a hmac-md5 -b <bit-length> -n HOST <key-file-name>
```

The default used by `rndc` to connect to is 953. Once the command is set up, the `rndc` command can be used with the following options shown in Table 9.1.

Table 9.1 `rndc` Options

`halt`	Stops named daemon immediately
`refresh`	Refreshes the database
`Reload`	Reloads the zone files but keep the cached files
`Stop`	Stops the service gracefully

EXERCISE 9.1: Making a Name Server Authoritative

In this exercise, you will learn how to make a secondary name server authoritative for a domain for a server running Bind 9 as you are about to decommission the existing primary name server.

1. In the *named.conf* file on the secondary server, check to see whether it is designated as the secondary for the zone you are interested in.
2. Change `type secondary` to `type master`.
3. Change the `allow update` option to be `none`.
4. Power down the master server and then reboot the server you have just changed to master.
5. This new server should now be authoritative for this domain. ■

Network Time Protocol

NTP is the defined protocol for synchronizing clocks of computer systems over a network, including the Internet. There are many time servers located across the world, and the source can be an atomic clock or via a satellite receiver. The accuracy that is obtained is very good and is typically no more than a few milliseconds. Like DNS servers, there is a hierarchy with primary or Stratum 1 timeservers and secondary or Stratum 2 servers. Those servers at a lower level always synchronize themselves to a server at a higher level. There are a number of public timeservers available, and those operated by the National Institute of Science and Technology (NIST) and US Naval Observatory (USNO) within the United States are particularly reliable.

Within the Linux system, there is an NTP client which is implemented as a continuously running daemon process. This process runs in the kernel space due to the sensitivity of timing.

The main configuration file for NTP is /etc/ntp.conf; and the servers you will synchronize to, as well as what networks are allowed to synchronize to your server, are described. You need to define a number of servers to synchronize to ensure there is redundancy in case of network or server failures. To specify a pair of servers to synchronize to, the ntp.conf will need to be modified as such:

```
server lerc-dns.lerc.nasa.gov # Stratem 1 server
server ntp.time.edu           # Stratem 2 server
```

You must then restrict the access that you allow these time servers; for instance, you do not want them to query your NTP server.

```
restrict lerc-dns.lerc.nasa.gov mask 255.255.255.255
   nomodify notrap noquery
restrict ntp.time.edu mask 255.255.255.255 nomodify notrap
   noquery
```

Using the same command, you can allow hosts on your local network to query your time servers.

```
restrict 10.10.10.0 mask 255.255.255.0 nomodify nomask
restrict 127.0.0.1
```

The use of the `restrict` command can seem to be confusing; but it should be thought of as everyone being allowed to query your server, and you are just reducing this overall access. Also note that in the code above, the local host (127.0.0.1) is allowed full access. You need to ensure that `ntpd` is started on boot with the `chkconfig` command, and also start the service manually for this first time with the `service` command.

```
chkconfig ntpd on
service ntpd start
```

To find and use an NTP server easily, there is a pool of servers that you can use. These NTP pools contain hundreds of public NTP servers that their operators allow to be used for time synchronization. You will need to define the pool you want to use with the DNS name *region.pool.ntp.org*, where you substitute region for the country or state where you live. More information can be found at www.pool.ntp.org. The utility program `ntpq` can be used to monitor the NTP daemon to determine the performance. The command uses standard NTP mode in six control message formats. This command is useful to see what time servers are currently being polled using `ntpq -p`.

In older versions of `ntp`, the local date and time can be set using `ntpdate`. This functionality is now in `ntpd`, but the older version is still in use. The command can be run manually, or usually it is run automatically at boot time.

`date` *Command*

The `date` command can be used to display or set the date on a system that does not have an NTP server installed. The options for the command are shown in Table 9.2, `date` options.

Table 9.2 `date` Options

`-a`	Adjust the date when the time has drifted.
`-u`	Display or set the time in GMT
`-s datetime`	Set the time and date

Windows Interoperability

There are numerous Microsoft Windows workstations and servers in use today, and interoperability with Linux is normally an essential task any system administrator needs to undertake. There a number of utilities to allow file sharing between systems, with the most common utility being *Samba.* Samba provides file and print services to all Server Message Block/Common Internet File System (SMB/CIFS) clients, which includes most versions of Microsoft Windows operating systems as well as Linux/UNIX servers and clients.

> **Note**
> SMB is an application layer network protocol, and is used as a shared-access protocol between different machines on a network. Microsoft modified the original specific for SMB and produced the CIFS, which was implemented in their products until Microsoft Vista was launched (which uses SMB 2.0).

Remote Desktop

With any mixed environment, you may need to mange both Linux and Microsoft Windows clients and servers. Natively within Microsoft Windows, there is a remote desktop application that uses the Remote Desktop Protocol (RDP). Within Linux, the open-source product `rdesktop` can be used to present the target desktop. This is based on the X-Windows system. To connect to a remote host http://hostname.mycorp.com with IP address 10.10.100.23, either of the following commands can be used:

```
rdesktop hostname.mycorp.com
redesktop 10.10.100.23
```

> **Note**
> The target server or client must have the remote desktop connection enabled for this to work. In addition, you may need to supply user name and password credentials applicable to the target host. The protocol runs on port 3389, which will need to be open on any intermediate firewalls.

Virtual Network Computing

VNC was developed by AT&T to administer machines. This is a client-server application, and there are now a number of different versions that are both open source and commercial. The VNC server itself is often built into the

Linux core, but may not be started. Once it is installed, start it by typing `service vncserver start`. If you are starting the server for the first time, you must type in a password at the prompt twice, which will be used by the remote client. The server is now started and ready to accept connection from a remote client.

There are a number of VNC clients available, and the one you use will depend on the company policy, the ease of use, and the level of security you wish to have. VNC operates on port 5901, and hence this needs to be opened up in any intermediate firewall (and possibly on your host firewall, if you have one). A client for X Windows will be vncviewer, which will connect to any VNC-compatible server.

Samba

Samba will implement the basic CIFS services, namely

- File and print services
- Authentication and authorization
- Name resolution
- Browsing or service announcement

Most end users will be interested in the file and print services; that is, the capability to share files between computers and to share printers. As with any system, a user may wish to share some of all their files and only allow certain users access (authentication and authorization). All of these are handled by the `smbd` daemon that is included within Samba.

The other daemon included with Samba is `nmbd`, which is basically name resolution on a point-to-point or broadcast basis. This daemon essentially is using the NetBIOS protocol to undertake the tasks. In broadcast mode, a client will send out a request to all machines on the network; for example, asking who is running a particular service. This may cause a lot of network traffic; but as it is confined to the local LAN, it is not usually an issue.

The other name resolution element of the `nmbd` daemon revolves around the NetBIOS Name Service (NBNS), or Microsoft's implementation of this, called Windows Internet Name Service (WINS). Within NBNS, there is a master NBNS server, which holds the IP address and NetBIOS name of each client or server on the network. The NBNS server will act like a normal nameserver; and when a client sends a request to it about a particular client name, the NBNS server will return an IP address if it is its database.

The browsing or service announcement part of Samba is also handled by the `nmbd` daemon. This should not be confused with Web browsing, but

browsing the network to see what file and printer shares are available on other computers.

There will be one Local Master Browser (LMB) on a network, and this is decided automatically by the nodes in the network. This LMB will hold the list of available services and provide these upon request (typically when a Windows client clicks on the **Network Neighborhood** button). In addition, in LMBs, these lists can be populated across domains via Domain Master Browsers (DMBs). Because of the time it takes to synchronize these DMBs, changes may take an hour or more to propagate across the domains.

Configuration Files

The main configuration file for Samba is smb.conf, which usually resides in either /etc/samba/smb.conf or /usr/local/samba/lib/smb.conf. This file can be edited manually, but there are many graphical user interfaces (GUIs) which are designed to make this easier, such as SWAT. The smb.conf file is different to many Linux configuration files in that its layout is similar to that used in older Microsoft Windows *.ini* files, comprising a number of sections with a section name in brackets ([]) delimitating the sections. The sections will contain information about the shares, printers, and services on the server. The correct terminology for the sections within Samba is *stanza*. There is one special stanza called `global`, which specifies parameters that apply to all other stanzas in the smb.conf file. A very minimal smb.conf file can be defined that just defines a couple of global parameters and some shares:

```
[global]
workgroup = mycorp
netbios name = computer_name
[share1]
path = /etc
comment = share the /etc folder to the world
[share2]
path = /documents
comment = share the global documents folder to the world
```

If you are setting up a server and want to share everyone's home directories, there is a special stanza called `homes`, which will enable the default home directory shares.

```
[homes]
comment = Home Directories
browseable = yes
Comment = only allow users to connect to their own
          directory, \\server\username
valid users = %S
```

```
comment = allow user to write to the directory
writable = yes
```

lmhosts File

The lmhosts file is built into Samba and is the NetBIOS name to IP address mapping, in a similar format to the /etc/hosts file. The file is located in the /etc/samba or /usr/local/samba/lib directories.

Managing a Samba Server

The Samba server has a number of daemons (notably `nmbd` and `smbd`) that need to be started. Once Samba is installed correctly, these will be started automatically upon boot and will read the smb.conf file described above. Once started, the server can be managed from the command line or through a GUI. The command-line interface is very easy to use, and the main command is `smbstatus`. The command can be issued with no options to display the full status of the servers and connected clients. In a large network, this will likely produce a large amount of output, so some of the options described below in Table 9.3 `smbstatus` options would likely be used.

Connecting to a Samba Server

As stated earlier, both Microsoft Windows and Linux clients can connect to a Samba server. Assume that there is a Samba server located on the server *syngress*, and this has shared the home directories of users as well as the *documents* share. If your username is rosie, you can map a drive on Windows to your home using the command line:

```
Net use h: \\syngress\rosie
```

You could also browse for the share within the *network neighborhood* option within Windows. If you have set up security on this share, you may be prompted to input your password to gain access.

Table 9.3 `smbstatus` Options

`-b`	Displays the list of users who are currently connected to the Samba server
`-s`	Displays the list of connected shares
`-L`	Displays the files that are currently locked
`-u username`	Displays information on the user *username*
`-p`	Displays a list of the smbd processes

> **Note**
> When you are prompted for your password when you try to connect to a Samba server, you need to enter your password that has been defined on that server. Depending on how the system has been set up, this may be different to your normal Windows password.

On a Linux or UNIX machine, there is a special client to access a Samba server called `smbclient`. The syntax for connecting to a Samba server is

```
smbclient //servername/sharename
```

This client, once properly connected, will display a new prompt to the user (typically `smb: \>`) and will have very similar functionality to an FTP session, where `get`, `put`, `ls` etc, can be used to navigate.

winbind

The integration of Linux and a Microsoft Windows can be time consuming, as there is no real underlying unified login. The `winbind` component of Samba tries to solve this problem by allowing the Windows domain user to appear and operate as a Linux user. The mappings between the Linux and Microsoft Windows user IDs are stored by `winbind`.

WEB SERVICES

The setting up of Web services on a Linux server should not be confused with just http access. The Web services will also include an FTP server, proxy server, and add-ons such as Java servers. Depending on the size and complexity of the network you are setting up, you will need one or more of these services, perhaps splitting the functionality across a number of servers. A brief description of each is needed to understand their interaction.

- A Web server will serve pages to the requestor using the HTTP.
- An FTP server will transfer data between a remote client and the server using the FTP.
- A proxy server will act as an intermediate server between a client and other networks, typically the Internet, to reduce the load on the connection.

The following sections will show how these are configured on a Linux system and how they can be accessed by the client.

Remote Access from the Command Line

While modern GUIs can be used to connect to remote systems, it is often easier and quicker to use a command-line utility. The common utilities that can be used are `telnet`, `curl`, and `wget`. `telnet` is a network protocol operating on port 23 and is a client-server protocol. The origin of `telnet` dates back to 1969 and was one of the first Internet standards. The security around `telnet` is not very great, as it does not encrypt any of the data over the connection and sends authentication data over the wire in clear text. The basic command is `telnet hostname|IP address`, which will connect to a `telnet` server at the specified *hostname* or *IP address.*

The other two utilities, `curl` and `wget`, are similar in that they retrieve files using HTTP, HTTPS, and FTP. Curl has more protocols available than `wget`. For using `wget` to download a URL, you can use the following code:

```
wget http://www.shell.com
```

The output from this is shown below:

```
$ wget www.shell.com
--2009-06-22 16:51:43-- http://www.shell.com/
Resolving www.shell.com... 134.146.83.23
Connecting to www.shell.com|134.146.83.23|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 26594 (26K) [text/html]
Saving to: 'index.html.1'

100%[=====================================================
============>] 26,594 86.4K/s in 0.3s

2009-06-22 16:51:44 (86.4 KB/s) - 'index.html.1'
    saved [26594/26594]
```

The utility can be used to download files using FTP with the following format:

```
wget ftp://www.syngress.com/*.jpg
```

For sites that have FTP usernames and passwords, these can be specified in the command line, such as `--ftp-user = user`. As the command is not interactive, it can be built into scripts to automate the process.

The other utility, `curl`, can transfer data using a wide range of protocols, including *telnet*. It was designed to work without user interaction to facilitate its use in scripts. It can be used to include the user authentication, the proxy support, and the resumption of data transfers. It is used in a similar method to `wget` above. The list of options is far too long to list here, and these can be seen using `man curl`. For instance, suppose you want to download the Webpage from www.shell.com.

```
curl http://www.shell.com
```

The first couple of lines of the output are shown below (highly truncated)

```
$ curl www.shell.com
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
    xml:lang="en" lang="en">
<head>
```

Apache (HTTP) and Tomcat

There are many different Web (HTTP) servers available for Linux, but the most common by far is the Apache Web server developed and maintained by the Apache Software Foundation, whose main Web page is located at www.apache.org/. The Apache Web server is probably the most popular Web server on the Internet today. While we will only look at the HTTP server, the foundation also has a large number of other projects that are continually being worked on, all of which are described on their home page.

The Apache HTTP server is an open-source server that has been developed for a number of operating systems including Linux, UNIX, and Microsoft Windows. While the server code is usually included with a Linux distribution, this will typically be out of date to some extent. If the server is going to be used to serve pages to the Internet, the latest version should always be downloaded from the http://apache.org Web site, which will include all the latest security fixes. The installation of Apache will be to /usr/local/apache2 as a default, but this can be configured by the user at install time. The installation will take up about 50 MB of disk space, which will include a number of options. Any user pages will increase the amount of disk space needed. The method of installation will be similar for other packages: download, extract, configure the makefile, compile, install, and make any additional configuration changes. The following sections will look at how you can configure the server to suit your specific needs.

Apache Configuration

The first configuration decision to be made is the location of the Apache server once it is installed. The default directory may not work for your particular system due to space constraints, or you may wish to segregate the code into separate directories for security reasons. If the server is going to be used to serve pages across the Internet, it would make sense to segregate these pages (and possibly the server itself) to add as another security layer. The actual directory where the server is installed is configured during the

`configure` task and is changed from the default using the `--prefix = PREFIX` option, where *PREFIX* will be defined as the installation directory. This directory is referred to as the *ServerRoot*.

apachectl

On a Linux system, the Apache server is run as the `httpd` daemon, which is often started on boot. There is a control script called `apachectl` that should be used to invoke the httpd daemon. This script will ensure that the correct environment variables are set before httpd is called. The script will pass any command-line arguments to httpd if necessary, or the script can be edited (if you want) to ensure that specific arguments are always passed to httpd. The control script syntax is `apachectl [http-argument]` and the main arguments are shown in Table 9.4, `apachectl` arguments.

Once you have started the Apache server, connecting to the server with a Web browser will display the default start page, as shown in Figure 9.1.

Table 9.4 `apachectl` Arguments

`start`	Starts the httpd daemon, displaying an error if it is already running
`stop`	Stops the httpd daemon
`restart`	Restarts the httpd daemon (or starts it if it is not running)
`status`	Displays a status report

FIGURE 9.1
Default Apache page.

httpd

Upon starting, httpd reads its configuration file httpd.conf, which by default is stored relative to the ServerRoot in conf/httpd.conf. Once it is started, it will create a pool of processes to handle all the requests that are generated at the server. The main options are shown in Table 9.5, `httpd` arguments.

Apache Modules

The Apache server can be customized with modules, of which many are available on the Web – particularly at http://modules.apache.org/, which is approaching 500 that are available for download. These modules can add a variety of functionalities to the Apache server such as bandwidth management, CGI, or authentication. The modules that are currently loaded can be listed using

```
apache2 -l  (shows those modules compiled in code)
apache2 -t -D DUMP_MODULES  (show all loaded modules)
```

Modules can be enabled and disabled using the `a2enmod` and `a2dismod` commands.

Apache Containers

Within the Apache configuration files are a number of containers, which are individual units that contain directives that alter the configuration of the Apache server. The two main containers are the *filesystem container* and the *Webspace container*. The filesystem container contains all the directives regarding the directories and files; for example, location and access rights. The Webspace container contains all the information about the Web site you are developing; for example, the URL name.

As an example, to enable directory indexes for the /var/web/dir directory, the following should be included:

```
<Directory /var/web/dir>
Options +Indexes
</Directory>
```

Table 9.5 `httpd` Arguments

`-d ServerRoot`	Sets the value for ServerRoot if different from the default
`-k start|stop|restart`	Starts or stops the daemon
`-v`	Displays the current version
`-X`	Runs in debug mode

The virtual hosts container can be used when you have multiple hosts being served from the same machine. This will allow different configuration options being applied to each virtual host. The virtual hosts can be IP-based (one IP per Web site) or name-based (multiple names on a single IP address).

Exam Warning

You should know that the Apache Web files are located in the directory specified by the DocumentRoot directive specified in the httpd.conf file.

.htaccess Files

The .htaccess file allows the Apache server to have a decentralized management of its Web tree. This file has directives in the plain text configuration file. These directives will apply to the directory where the .htaccess file resides and is read upon every access; so any changes will have an immediate effect. If the .htaccess file has options in it, the Apache configuration file must be configured with the `AllowOverride Options` set.

EXERCISE 9.2: Stopping and Starting the Apache HTTP Server

In this exercise, you will learn how to stop and start the Apache HTTP server daemon. This is often needed if you change configuration details or add some functionality. A number of these changes are often only read when the Apache server starts.

1. You wish to stop the Apache server daemon gracefully (allowing processes to finish if they have time), so you need to execute the command `apachectl -k graceful-stop`.
2. When all the processes have stopped, you can restart the server using the command `apachectl -k restart`. ■

PHP

PHP is a general purpose scripting language used in Web development. It is mainly used when you need to have a dynamic Web page created and, once created, appears just like any other HTML page. It is often installed as part of a LAMP environment (Linux, Apache, MySQL, and PHP); with a good example of this being XAMMP, which can be downloaded at www.apachefriends.org/en/xampp.html. When PHP starts, it reads its configuration file php.ini, which is usually in `/usr/local/lib/php`. When

it is run as an Apache module, the configuration directives will reside in memory until a new Apache process has started.

CGI Scripts

Apache can be configured to treat any file in a particular directory as a CGI script. This is typically referred to as the *cgi-bin* directory by Web developers. The specific directory where the default CGI scripts are held can be set up in the httpd.conf file using `ScriptAlias`. The syntax of this is as follows:

```
ScriptAlias URL path name
```

If you have set up your CGI scripts in a directory /usr/local/apache2/CGI-bin/, then to make the Apache server treat all these files in this directory as a CGI scripts, the full command will be:

```
ScriptAlias /cgi-bin/ /usr/local/apache2/cgi-bin/
```

It is possible to run CGI scripts from any directory as well as that defined using ScriptAlias. For this to happen, there are two distinct steps that have to be undertaken: First, the *cgi*-script handler must activate AddHandler directive; and second, the ExecCGI directive must be specified in the Options directive. The httpd.conf file will need to be modified by adding the line

```
AddHandler cgi-script .cgi
```

Under the directory container, the `options` line will be added:

```
Options +ExecCGI
```

As an example, suppose you want to allow CGI programs to be located and executed from a file ending in *.cgi*; the following should be added:

```
<Directory /home/*/public_html>
 Options +ExecCGI
 AddHandler cgi-script .cgi
</Directory>
```

Configuring Apache Server Logs

The Apache error log is set by one of the directives, `ErrorLog`, and is the log file where the Apache httpd daemon will send all the errors and diagnostic information to, which is normally called error_log. Because of this, it should be the first place you look when you are diagnosing an Apache server problem. It will also include any error messages and debug information from CGI scripts. The `LogLevel` directive in the configuration file will define the amount of error logs, or how verbose they are. There are eight levels of logs: `emergency`, `alert`, `critical`, `error`, `warning`, `notice`, `info`, `debug`.

When you first start your Apache server, you may wish to have more verbose logging (that is, debug) until you are confident that everything is working correctly.

Exam Warning

Do not allow anyone to write to the Apache log directory, as this will almost certainly give them access to the uid that the server is started, which is often root. In addition, the raw logs can have control characters inserted into it by malicious users, so care must be taken when viewing the raw files.

Tomcat Configuration

Tomcat (or its full name, Apache Tomcat) is another product developed under the Apache license, with the main Web page being http://tomcat.apache.org. It is a servlet and JavaServer Pages (JSP) container for the Apache Web server. JavaServer and servlet specifications are defined fully in http://java.sun.com/products/jsp/download.html and http://java.sun.com/products/servlet/download.html, respectively. JSP allows developers to create dynamically generated Web pages using HTML and XML and deliver these to a Web client. Servlets are the Java equivalent to PHP, CGI, and ASP.NET and can be automatically generated by a JSP compiler. These two technologies are therefore linked together. The Java Development Kit (JDK) must be installed and should be working on the system before you install Tomcat.

The layout of the Web application can be considered to be the hierarchy of directories and files. This can be packed into a form known as a *Web Archive*, or WAR file. The topmost directory in this structure is known as the *document root* and all files can be referenced from here. There is a standard to a WAR file, with the following files in or referenced from the document root:

- ∗.html and ∗.jsp (HTML and JSP) pages are stored in the document root or (for larger applications) into a subdirectory hierarchy.
- /WEB-INF/web.xml is the *Web Application Deployment Descriptor* for your application and is an XML file describing the components that make up the application.
- /WEB-INF/classes/ is a directory containing Java class files for your application.
- /WEB-INF/lib is the directory containing the JAR files and third-party libraries.

File Transfer Protocol

The FTP server is used to upload and download files to a server from an FTP client. The protocol itself dates back to the early days of UNIX and is a true client-server architecture. The client interface has evolved from the original command-line mode to sophisticated GUIs in most operating systems today. The standard Web browser, such as Firefox or Internet Explorer, can be used as a front end (substitute http:// with ftp://). The protocol can be used with or without usernames and passwords, although security is still an issue due to the transmitting of these credentials in plain text. An FTP server that allows anyone to connect to it is called an *anonymous server.*

The server can be set up in one of two modes: *active* or *passive*. In active mode, the server listens on port 21 for incoming connections, and this is used as the control stream. The server binds port 20 as the data connection to the client. In passive mode, the data port is on an arbitrary high port. The different modes often lead to confusion, so the two modes will be explained in detail below.

Active FTP

In active mode, the FTP client will connect from a random unprivileged port P > greater than 1023 to the FTP server's command port 21. The client will start to listen on port P+1 and sends this information to the FTP server using the command *PORT P+1.* The FTP server will then connect to the P+1 port from the FTP server from its data port, port 20. On the client side, therefore there will be two open ports, P and P+1, and on the server side ports 20 and 21 will be opened and connected back to P and P+1 on the client.

Passive FTP

In active FTP, the server initiates the connection to the client, which often causes issues if there are firewalls in the way. Passive FTP was therefore developed, and the client initiates passive mode using the command `PASV`. In this method, the client initiates both connections, which alleviates any firewall issues. In this case, the FTP client opens two unprivileged ports P and P+1, with P again a port above 1023. The first port will connect as before on port 21 for the FTP commands, and then issue the `PASV` command instead of the `PORT` command. The FTP server then opens a random unprivileged port above 1023, say S, and then sends the `PORT S` command back to the client. The client will initiate the connection to this port on the FTP server.

The setup of an FTP server is relatively easy for the base installation. As security is an issue with the FTP protocol, you may wish to install a secure file transfer server such as SSH. Most modern Linux distributions will have

the SSH server as an option. The main decision to make is whether to install an authenticated or anonymous server. Unless you have a very trivial setup and no Internet access, it is recommended that an authenticated server is used. Once installed, a user can connect to your server using the IP address, or DNS name using the command line or GUI. If the unsecured version has been installed, the user may be able to traverse across the server and upload and download files to/from many locations.

An FTP server can be set up to restrict the local usernames that can be used. For instance, you should typically stop remote users using the *root*, *bin*, etc username. This is achieved by putting the list of users in the /etc/ftpusers file. This simple list of names must also appear in the /etc/passwd file. The /etc/ftpchroot file is in a similar format to the /etc/ftpusers file and contains the list of users whose session root directory needs to be changed. This is usually to the directory listed in the /etc/ftpd.conf configuration file. This is to ensure that the remote user is put in a safe directory and they cannot traverse into an area where there are sensitive files.

EXERCISE 9.3: Transferring Files in ASCII or Binary

In this exercise, you will transfer an executable file from a server to a host with a command line `ftp` utility.

1. First, connect to the FTP server using `ftp servername`, where *servername* is the name or IP address of the server you wish to connect to.
2. When you are prompted for a username and password, enter the valid credentials.
3. When the *ftp>* (or similar) prompt is displayed, type `binary`. This will change the transfer mode to binary from ASCII (or text transfer)
4. Type `get filename` to retrieve the file you wish.
5. The file is now transferred to your computer in the correct format. If you had used the ASCII mode, the file would have been corrupted and unusable. ■

Squid

Squid is a proxy server and Web caching service. There are a number of uses for it, and they are as follows:

- Caching Web lookups for the network
- Additional security for Web access
- Speeding up a Web server through caching of repeated requests.

The main protocols that are used with Squid are HTTP and FTP, although there is some support for others such as SSL. When Squid is installed on the local network, it can be configured to cache HTTP and FTP traffic that is destined for the Internet, which can reduce the amount of traffic on the Internet gateway. Web clients are configured to access to the Internet through this cache, and if a firewall is installed, this is usually configured to block direct Internet access from those clients. As Squid is intended to increase the user's experience, it is essential that it is installed on a system which is powerful enough to cater for all the requests it is expected to receive. Fast hard disks to retrieve the files from cache are therefore recommended.

Squid Configuration Files

Squid itself should be configured to run at boot time and will be a normal Linux daemon. The main configuration file for Squid is /etc/squid/squid.conf and the software is installed in /usr/local/squid. The main options that are likely to be configured will be described. Web servers normally listen on port 80, but Squid can be configured to listen on any port. This is often useful if you are trying to hide some servers from general browsing. The default port for Squid is 3128, although a lot of people change this to port 80 to make it easier to remember. To change the port in the squid.conf file, you will need to add a line such as:

```
http_port 3128 80 8080
```

This will allow Squid to listen on ports 3128, 80, and 8080. Depending on the installation of this server, you may need to add multiple ports to ease the configuration changes needed on the client machines. For example, some proxy servers will be configured to run on port 80, and adding the port 80 to the Squid proxy configuration will greatly ease the transition.

In addition, you need to give the system a valid DNS entry for the system, which will ease the configuration now and in the future. Having a generic name such as *proxyserver.mycorp.com* will allow you to change the system at a future date, even its IP address and still give clients the ability to connect to it without having to re-configure all of their Web browsers.

The security on the server can be added in layers. The first layer can be to use simple access control lists (ACLs), which can restrict the networks that are allowed to connect to the server and with protocol. If your local network is a subnetted class A network, 10.10.100.0/24, and you wish to allow this entire network to access the Squid server using HTTP and deny all other networks, the following lines should appear in the squid.conf file:

```
acl mynetwork src 10.10.100.0/255.255.255.0
http_access allow mynetwork
http_access deny all
```

The Squid proxy can also be used to filter individual sites so that users cannot access them. Suppose you do not want your users to access CNN.com or bbc.co.uk. The following lines will deny these sites:

```
acl newssites dstdomain .bbc.co.uk .cnn.com
http_access deny newssites
```

You can be more specific about when you allow or deny access to a site; for instance, you may allow the users to access the domains in the "newssites" group specified above during the lunch hour. For this, add a new acl, and the configuration file should look like that given below:

```
acl newssites dstdomain .bbc.co.uk .cnn.com
acl lunchtime MTWHFAS 12:00 13:00
http access allow newssites lunchtime
http_access deny newssites
```

Note that the specification of the days of the week is Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, and Sunday. When Squid matches a line in the configuration file for http access, it will execute this and stop reading the file. The order of the lines is therefore important. Obviously, the more rules you require, the more complex the file will become. Often, used rules should therefore come at the beginning of the file whenever possible, to speed up execution. You can also block certain IP addresses or groups of addresses or allow full access for certain IP addresses, such as servers.

Exam Warning

Squid can be used as a proxy server and as a caching server. When it is a caching server, it displays data to the client that it has already in its cache; for example, a Web page. When it is used as a proxy, it is acting as a go-between for requests from a client to the target server. The target server will therefore not be able to communicate directly with the client.

APPLICATION SERVICES

The final section in this chapter will look at the application services that can be installed and run on a server. The main application services that are required by most companies are examined – namely printing, mail, and a database. These application services should not be confused with the client's view of these, who can have a local printer and an electronic mail (e-mail)

client, for instance. The application services described here will serve up the application to many people within an organization and allow for central management.

For large companies, the number of printers can be very large on a single floor of a building, let alone the whole company. The management of these from one or more central locations will often be a requirement to reduce this service overhead. The setting up of the CUPS print server and the basic administration of this server is defined. The CUPS server provides the capability to pool printers together such that a user can execute the print command and the least busy printer will be used. This will allow printer administrators to take printers offline for maintenance without impacting the user.

As a company grows, the use of a central mail server becomes sensible to better control the flow of data into and out of the company, to allow the use of a central antivirus and spam checker, and to allow the management of a user's mail at this central location. The two main servers for Linux (sendmail and postfix) are discussed and the basic configuration of each is described.

Finally, the MySQL database application is described. While many will think that a database is usually a client-side application, MySQL can often be used on the server as part of another application (for instance, the postfix mail server can use MySQL as its database). The MySQL database server is the most popular open-source database around and is usually found on most Linux distributions.

Printing

In Chapter 8, there was a section on printing that related more to the client and how a user can manage their printing. This section also detailed the print commands such as `lpr`, `lp` that can be used from a command line. If you have not read that section yet, you should do so before reading this section, as it will teach you about the tools needed to test the print server when you have set it up. The print server that is common within Linux is CUPS, or the Common UNIX Printing System, and the latest version can be downloaded from http://cups.org (along with all the documentation you will need).

The CUPS server can manage multiple printers, both local to it and remotely across the network. The server itself converts the page descriptions from the application you are trying to print from data-specific to the printer you are trying to print on. Each printer will have differences in the format of data it requires, depending upon different manufacturers or even different models within one manufacturer's range. CUPS will keep track of the

printers it is managing and display messages when a printer needs attention, perhaps due to no paper or an ink cartridge that needs replacing.

The initial task to set up CUPS is to add one or more printers to it, which is done under the *Administration* tab of the CUPS interface. This interface is accessed via a Web browser using port 631. To access CUPS that is running on your local machine, you need to type http://localhost:631, and you can substitute the *localhost* part with a machine name or IP address if you are accessing a remote CUPS instance. There will be a number of pages within the interface that will require the root username and password to be entered, adding a printer being one of them.

Network Printers

Network printers can be added to and managed by CUPS. The addition of network printers is very similar to that of a local printer, but the remote IP address (or name) must be known or found using Simple Network Management Protocol (SNMP) built into CUPS. There are three network protocols supported by CUPS:

- AppSocket Protocol, usually associated with HP JetDirect network interfaces.
- Internet Printing Protocol (IPP), normally over port 631
- Line Printer Daemon (LPD) Protocol, which should only be used if the above protocols are not supported. The port associated with LPD is normally 515.

Managing Operation Policies

There are rules built into CUPS to allow the administrator to define a number of policies, such as the user must supply a password. This allows the administrator to customize the interface to match the policies within the company. These rules are stored in the *cupsd.conf.* These rules are easily changed via the CUPS interface using the *Edit Configuration File* on the **Administration** tab.

Printer Classes

Within CUPS, an administrator can group printers together to form a class. This allows the user to send a document to this group of printers, and CUPS will decide which printer is idle and then print the document on this printer.

EXERCISE 9.4: Allow Users to Cancel Any Job

You want to allow all your users the ability to cancel any job on a printer, not just the one they own.

1. Start the CUPS interface by typing the URL http://localhost:631 into a browser.
2. Click on the **Administration** tab.
3. Under the *Basic Server Settings*, make sure that the option *Allow users to cancel any job (not just their own)* is checked.
4. Click on **change settings**.
5. When prompted, type the **super username** and **password**.
6. The CUPS server will restart, and then any user can cancel a job. ■

Mail

The e-mail server will be an important part of any network, and very useful in a small or home office environment. A distinction should be made between the e-mail server and client – a well-configured e-mail server will be able to have a number of different clients seamlessly connecting to it. This will be very useful in organization where there are mixed systems (Linux, Microsoft Windows, Apple Macs, and so forth). This section will only touch on the clients where it is necessary to see the application working, and will not concern itself with their configuration.

This chapter describes the setting up of the `sendmail` mail transport system; however, there are many others available for Linux, such as postfix, which are very popular. Learning the basics with one will, however, give you a very good advice on how to set up different servers. The first step in setting up a mail server for a domain does not involve the server at all. Initially, the DNS for your domain needs to be configured to ensure that there is a valid MX record pointing at your mail server (or where it will be).

The main transport mechanism for e-mail around the Internet and between e-mail servers is the Simple Mail Transfer Protocol (SMTP), which operates on port 25. The basic protocol has been around a number of years, although a number of extensions have been added for authentication and error reporting, among others. The mail is transferred from one e-mail server to another using a Mail Transfer Agent (MTA); sendmail, postfix, and others are classed as MTAs.

The Post Office Protocol (POP) and Internet Message Access Protocol (IMAP) protocols can be used by the local e-mail clients to retrieve their e-mail from a remote server. Currently, these are the two most prevalent protocols for e-mail retrieval. These can be used to connect to your local server, as well as Web-based mail servers such as Gmail. POP3 is the latest approved standard release of POP, and has addressed some of the security

issues that were inherent in the original releases. POP4 has been developed, but is not an approved standard at the time of writing. However, it is still not totally secure, and care should be taken when using it across an unsecured network.

The IMAP protocol has gained in popularity over the years, and is now offered by Web providers such as Gmail (as well as POP3) and for local servers such as Microsoft Exchange. IMAP will be much faster to download mail on a local network compared with POP3.

How Mail Works

Any mail server in your organization will ideally need to handle both incoming and outgoing mails. The mail arrives at your mail server for the domain(s) it looks after, defined in part by the DNS MX record. Users within this domain will have usernames associated with their mail account. While you may find it useful to give people their first name like Fred and Mary, this will not work when the company grows larger. It is therefore useful to define a scheme before setting up the server, perhaps first.lastname (or whatever works for your organization). The full e-mail address will then be *username@your_domain*. E-mail will then be stored on the server until the user successfully retrieves them using an e-mail client.

Outgoing mail will be handled slightly differently. If a user sends an e-mail destined for a local user, the mail server will simply put the e-mail in the appropriate user's mailbox, where they can retrieve it as above. If it is for someone outside the domain, the mail server will look up the MX record of the target domain and then try to send or `relay` the mail message to that server.

Sendmail

Sendmail has been around for a number of years and has grown in complexity to keep up with the demands of users in requiring more sophisticated mail. The following sections will outline how to set up a sendmail server.

Starting and Stopping Sendmail

Sendmail is usually started upon boot, but can also be started and stopped afterwards. This is especially important when configuration changes are made, as sendmail will only read the configuration file once when it starts. The commands below are used to start, stop, and restart sendmail.

```
service sendmail start
service sendmail stop
service sendmail restart
```

Sendmail Configuration

The basics of sendmail configuration will be given here, but it should be noted that `sendmail` is very complex with whole books devoted to the correct set up. Mistakes in the sendmail configuration file can cause sendmail to stop processing mail, so performing a backup is advisable. The main sendmail configuration file is sendmail.cf, which is now normally located in /etc/mail/sendmail.cf (and in /etc/sendmail.cf in some older versions of Linux). This file can be edited directly, but the syntax can be confusing and, as stated above, mistakes may stop the server from working.

A common method of producing the sendmail.cf file is through use of the *m*4 macro processor, which works on the configuration parameters in the file /etc/mail/sendmail.mc. While a lot easier than editing the sendmail.cf file directly, it is by no means simple. Methodical working through the file and observing the logs when you start sendmail are recommended. As the complete list of commands is large, we will concentrate on a few often-used ones.

Smart Hosts

While the relative cost of an always-on Internet connection, such as cable or leased line, has reduced in the past few years, this is favored by a lot of companies. However, this is not the case in many parts of the world. In addition, what happens if you are trying to send e-mail to a company who has their e-mail server turned off? In these cases, you can configure `sendmail` to pass mail onto another sender rather than delivering it directly. In the sendmail.mc file, the following need to be added:

```
dnl # define the Smart Host
define('SMART_HOST', 'smtp.smarthost.org')
```

In the above, the line starting `dnl` means *do not list*, and basically tells the m4 macro processor to put the rest of the line in as a comment. The define section will tell sendmail to pass its mail to the server smtp.smarthost.org for onward delivery.

Mail Delivery Intervals

You may wish sendmail to deliver mail at certain times as opposed to it trying to deliver the e-mail instantly it receives it, as the default is. This may be useful to reduce the load on the network, or if you do not have an always-on connection. The sendmail.mc file should be edited as follows.

```
dnl # define sendmail to delay delivery
define('confDELIVERY_MODE, 'd')
```

Building the sendmail.cf file with m4

Once you have built the sendmail.mc file, you must run it through the m4 macro processor to generate the sendmail.cf file.

```
m4 /etc/mail/sendmail.mc > /etc/mail/sendmail.cf
```

Mail Relaying

Setting up mail relaying is an essential task that needs to be undertaken to ensure that your server is not used by spammers. Basically, you want to relay (or deliver) all mail that originates from your domain to the target domain. If your domain is http://mycorp.com, then you need to add this into the /etc/mail/relay-domains file. As it is relatively easy to spoof the from address, you need to be more specific on who can use your mail server. The configuration file /etc/mail/access can be more specific on who can use the relay server. This file can be used to configure a number of actions such as *RELAY*, *REJECT*, and *DISCARD*. The file is a simple 2 column list, which needs to be converted into a sendmail-compatible file. A sample file is shown below:

```
localhost    RELAY
10.10.100.12 RELAY
10.10.100.13 RELAY
mycorp.com   RELAY
```

This will be converted into a sendmail format file named /etc/mail/access.db:

```
cd /etc/mail
make
```

Postfix

An alternative to sendmail is *Postfix*, which was designed to be simpler to configure than sendmail. Although there are several hundred configuration parameters that are controlled by the main configuration file /etc/postfix/main.cf., the variables in the postfix.cf file are defined and used in a similar way to shell variables.

```
parameter = value
new_parameter =$parameter
```

The domain that is used in outbound mail is defined in the *myorigin* parameter, which by default defaults to the local machine name. You obviously need to change this (unless you are setting up the server for a small site, such as a home office). The server can accept mail for a number of domains, and these will be specified in the `mydestination` parameter. As with sendmail, you must ensure you only relay mail from hosts or network you know and trust. This is undertaken using the `mynetworks` parameter.

As an example, to permit relaying from your local network, you may specify *mynetworks* as follows:

```
Mynetworks = 10.10.100.0/24
```

The `postfix` daemon will report all errors to the syslog daemon, which itself sorts out events by class and severity. The logging classes, levels, and logfile names must be entered into /etc/syslog.conf to ensure these are logged correctly.

E-mail Aliases

The aliases file allows you to have a number of valid e-mail addresses that do not have a specific user account. In a typical system, the majority of these aliases are configured to root. The file will normally be in /etc/aliases or /etc/mail/aliases. This file can be used to set up simple mailing lists.

```
# System aliases that must be present
mailer-daemon:        postmaster
postmaster            root

# Pseudo accounts
bin:                  root
abuse:                root

# Mailing lists
# Senior managers
managers:            dave,john,anne
```

You may not wish to store the mailing list in this file, as you can then prevent users changing this base file. A better way is to store the list in a generic directory and then include it in this file, as below:

```
# Mailing list for the managers
managers-list:     "include:/home/mail/managers-list"
```

The aliases file will need to be converted to a format that can be parsed by the mail server. This is achieved using the command `newaliases`, which will convert the file into a random access database. The command is identical to the command `sendmail -bi`.

MySQL

The MySQL application is a database server that is included or can be downloaded for most Linux systems, as well as for Microsoft Windows systems. MySQL is a relational database management system (RDBMS), as are others such as Oracle, Microsoft SQL server, and DB2. An RDBMS will store data in

tables, which can be linked together to enable manipulation of the data much easier. As an example, suppose you wish to catalog your music collection into a database. You may wish to set up a database with a number of tables: you could put the artists in one table; album title, year in another; and add tracks in another. The use of MySQL is not in the scope of the exam, and we will only deal with the basics of its configuration. This is often important, as a number of applications require a database to be properly installed for their application to be installed and run.

MySQL Configuration

The main configuration file for MySQL is located in /etc/my.cnf, and the MySQL databases are located in a subdirectory of */var/lib/mysql*. Hence, for the music database above, you may want to create a database called music, and this would be created in */var/lib/mysql/music.* The my.cnf file does not usually need editing manually and is usually only done to fine-tune the application. It is not necessary that the MySQL data directory should be owned by the user which runs MySQL, and this directory should be set to 700 using `chown`.

There is another configuration change that should be made when you first install MySQL. The out-of-the-box installation does not set a password for the MySQL root user, and this can be set using the `mysqladmin` command:

```
mysqladmin -u password newpassword
```

Starting and Stopping MySQL

The MySQL service is usually started on boot, as it needs to be running before you can create or access any databases. It can be started and stopped using the command line using one of the commands listed below.

```
service mysqld start
service mysqld stop
service mysqld restart
```

Testing the Connection

The MySQL server can be tested very easily using the in-built command-line interpreter. This is a necessary tool for basic administration and can be used to ensure that the server is working correctly. The basic command is `mysql`, and the options are shown in Table 9.6.

While the password can be entered on the command-line interface, it is not recommended due to security concerns. If MySQL is running and the correct credentials are entered, the system will respond as follows:

Table 9.6 `mysql` Command-Line Options

`-u username`	Connect to the database a *username*
`-p`	Prompt for password
`-h hostname`	Connects to the MySQL server on the remote host *hostname*

```
# mysql -u root -P
mysql>
```

When you are attempting to connect to a database that is on a remote host, `mysql` will attempt to connect to that database server using port 3306. This port must therefore be open between the two systems, and may require suitable rules put into any intermediate firewall.

Learn by Example: Changing the mysqladmin Password

One of the administrators of your system has left, and they had the MySQL administrator's password. You need to change this to ensure the security of your application. The old password is *ABC123*, and you want to make the new password *123ABC* (make them much more difficult in real life!). Type the command:

```
mysqladmin -u root 'ABC123' password '123ABC'
```

The root password has now been changed.

SUMMARY OF EXAM OBJECTIVES

In this chapter, you learned about how to configure a Linux system when it is used as a server. The three distinct sections were the network, Web, and application services, and each was split up with specific services essential to Linux administrators. The initial section was centered on the network services, which contains a number of services that are essential to the network, and which end-users often take for granted. The DHCP service provides the client with an IP address and other data, such as the local nameserver. The allocation of IP addresses to clients is vital to ensure that there are no address conflicts on the network. In addition, the DHCP server can provide the address of the local nameserver to ensure that the name resolution can occur. The basics of DNS configuration were discussed, and how to set up the different files for forward and reverse name resolution was outlined. The interoperability with Microsoft Windows using a Samba server was defined,

and how to modify the various configuration files to undertake this task was described. Finally, in this section, the use of an NTP server was defined, and how this can be set up to serve an accurate time signal to clients on the network was described.

The section on Web services was centered around the Apache Web server, which is a very common Web server (both on Linux and other platforms). How the Apache server is configured was discussed, along with where the main configuration files are located. In addition, the definitions of modules and containers pertinent to the Apache configuration were discussed. How PHP and CGI scripts are incorporated into the Apache server was described. Command-line access to servers (using commands such as `curl` and `wget`) was shown, and the typical output that was obtainable was shown. The Squid proxy server configuration was shown, and how it is used in a network was described. Finally, the configuration of an FTP server, and how this could be configured for different file transfers (such as straight ASCI text or programs in a binary form) were described.

SELF TEST

1. A DHCP server can be used to set up the following on a client:
   - **A.** Fixed IP addresses and DNS zone data
   - **B.** Routing and leased IP addresses
   - **C.** Leased IP address and default printer IP address
   - **D.** E-mail address of the user
2. A DNS has just been set up in your company. The primary purpose of this server is to
   - **A.** Enable Web browsing to occur.
   - **B.** Translate domain names to IP addresses.
   - **C.** Act as a gateway for users who wish to browse the Internet.
   - **D.** Act as a file and print server for Microsoft Windows client.
3. You have just installed an NTP server onto your computer and want to set up a number of time servers in the configuration file. If you performed a standard installation, what file do you need to edit?
   - **A.** /etc/ntp.conf
   - **B.** /etc/ntpd.conf
   - **C.** /sys/ntpd.conf
   - **D.** /bin/ntp.conf

4. You have just installed an Apache 2 Web server onto a server. The installation was successful and you now wish to start the server. What command would best accomplish this?
   - **A.** `apachectl start`
   - **B.** `apacheweb start`
   - **C.** `apache start`
   - **D.** `httpd -k start`

5. Your company has installed a MySQL server on your network. You have used traceroute to confirm that you have network connectivity to that server. Which port would your client program use to connect to the MySQL server?
   - **A.** TCP port 631
   - **B.** UDP port 3306
   - **C.** TCP port 3306
   - **D.** TCP port 631 and UDP port 3306

6. You are administrating a Samba server on your network. You want each user to connect to his/her own home directory. What configuration changes would you need?
   - **A.** Add the line `valid users = %S` in the smb.conf file
   - **B.** Add the line `brosweable = yes` in the smb.conf file
   - **C.** Add the line `home = %S` in the smb.conf file
   - **D.** Add the line `path = ~` in the smb.conf file

7. You have just added a DHCP server onto your network to reduce the network administration tasks you have. Before you turn on DHCP on each of the clients, you want to test the connection. Which is the best command to achieve this?
   - **A.** `dhcpd`
   - **B.** `bootp`
   - **C.** `dhcpcd`
   - **D.** `pumpd`

8. You want to install a proxy server in your network and have chosen the Squid proxy. You have a mixed network of Linux, Sun Solaris, and Microsoft XP clients. You want to keep the default Squid port but allow for migration of existing clients. What would be the best setting for the squid.conf file?
   - **A.** `http_port 80`
   - **B.** `http_port 3128 8080`
   - **C.** `http_port 3128`
   - **D.** `http_port 3128 80`

9. In your DHCP server, you wish to allocate a fixed IP address to one of your color laser printers. What do you need to do to set this up?

A. Find the IP address the printer is currently assigned and fix it using the `dhcpcd fix IPaddress` command on your DHCP server.

B. Ensure *bootp* is available on the printer and then assign an IP address to this in the dhcpd.conf file with the host parameter.

C. Find the printers' MAC address and a spare IP address from the pool of IP addresses defined on your DHCP server, and set this up in the dhcpd.conf file with the host parameter.

D. Set a fixed IP address directly on the printer and allow this to broadcast it to the DHCP server upon the printer being started.

10. Your new Apache Web server has been set up and one of the developers wants to know which directory to load the Web pages. How can you find out which directory this is?

A. Look in the httpd.conf file for a `WebRoot` directive.

B. Look in the httpd.conf file for a `DocumentRoot` directive.

C. Look in the httpd.conf file for a `ServerName` directive.

D. Look in the httpd.conf file for a `WebBase` directive.

11. You have been told that your mail MTA is being used as a relay by spammers. You want to stop this happening. What is the best course of action?

A. Relocate the mailserver behind your corporate firewall and only allow TCP port 25 to and from this server.

B. Ensure that the only hosts that the mailserver will allow to relay are on your local network by configuring the /etc/mail/access file.

C. Ensure that only your domain can be relayed by configuring the *relay-domains* file.

D. Configure the mailserver to stop all relaying of mail and make sure all the users connect to it via an approved client.

12. The speed of your Internet connection has slowed down because of the increase in the number of employees in your company. You have installed a Squid proxy and now wish to restrict the browsing of certain sites to lunchtimes only. You have set the list of banned sites up as an acl called banned_sites. Which is the correct configuration in the squid.conf file?

A.
```
acl lunchtime MTWTFSS 12:00 13:00
http_access allow banned_sites lunchtime
```

B.
```
acl lunchtime MTWTFSS 12:00 13:00
http_access allow banned_sites lunchtime
http_access deny banned_sites NOT lunchtime
```

C.
```
acl lunchtime MTWHFAS 12:00 13:00
http_access allow banned_sites lunchtime
http_access deny banned_sites
```

D.
```
acl lunchtime MTWTFSS 12:00 13:00
http_access allow banned_sites lunchtime
http_deny banned_sites
```

13. You want to set up your Apache server to capture logs, as you are having problem with the application. What log level would you set to give you the most verbose logs?

A. `emerg`

B. `error`

C. `info`

D. `alert`

14. The Samba server in your office has been set up with the name *samserv*. You want to connect to the sammy directory that has been set up and shared on it. What will be the correct command from a terminal shell if you want to connect as a user called juliet?

A. `smbclient //samserv/sammy juliet`

B. `smbclient //samserv/sammy -u juliet`

C. `smbclient //samserv/sammy -U juliet`

D. `smbclient //samserv/sammy U juliet`

15. You want to administer your DNS using the `rndc` command, but you cannot connect to the server. You have pinged the server and it responds. You have just installed the client on your machine. What is the likely error?

A. You have not put your machines' IP address into the rndc.conf file for the target DNS.

B. You have not inserted the correct keys into the rdnc.conf file.

C. You have run the `dnssec-keygen` command immediately before issuing the `rndc` command.

D. You must run the `dnssync` command on the new host and the target to ensure they can communicate with each other.

SELF TEST QUICK ANSWER KEY

1. B
2. B
3. A
4. A
5. C
6. A
7. C
8. D
9. C
10. B
11. B
12. C
13. C
14. C
15. B

CHAPTER 10

Securing Linux

Exam objectives in this chapter

- Managing and Monitoring User and Group Accounts
- File Permissions and Ownership
- SELinux Basics
- Implementing Privilege Escalation
- Security Applications and Utilities
- Checksum and File Verification Utilities
- Implementing Remote Access
- Authentication Methods

UNIQUE TERMS AND DEFINITIONS

- **Sandbox** A protected, limited environment where applications (for example, Java programs downloaded from the Internet) are allowed to "play" without risking damage to the rest of the system.
- **Mandatory Access Control** MAC is a type of access control where system privileges are specified by the system. They cannot be applied, modified, or removed – except perhaps by means of a privileged operation.

- **Discretionary Access Control** DAC is a type of access control where system privileges are specified by the owner of an object, who can apply, modify, or remove them at will.

INTRODUCTION

Linux is regarded as a very secure operating system. However, even the most secure systems can have an occasional flaw or be misconfigured. Even once you get everything set up and buttoned down, keep in mind that, as security expert Bruce Schneier says, "Security is a process, not a product."[1] Additionally, the security experts at the SANS institute recommend what they call "defense in depth,"[2] meaning security is best applied in layers. In this chapter, we'll look at the tasks necessary to make sure that your Linux systems live up to their secure reputation.

MANAGING AND MONITORING USER AND GROUP ACCOUNTS

You may be wondering "Why even have separate user accounts?" On a home computer, it can be convenient to use different accounts so each user can keep things just the way they want without conflicting with other users. In a corporate environment, it's critical to be able to control who gets access to what information to maintain security and may even be a legal requirement to protect privacy. It is also handy to know who is doing what on a computer to assist in troubleshooting problems, and unfortunately, it is occasionally necessary for tracking inappropriate activity. Finally, not only is it important to limit user accounts to protect information but it also protects the system itself from both malice and simple errors. Ideally, even the administrator should use full administrative access only when necessary to minimize those little "oops" moments.

When you have a handful of users connecting to a computer, the job of managing individual sets of permissions is not that onerous. Once the community of users grows beyond a handful, this job can quickly grow out of hand. Imagine the task of managing individual users' permissions on a file server that has hundreds of users connecting to it. Your life would take on all new meaning, and it would not be in a positive direction. Assembling users into groups make this manageable because you can manage permissions for the group. When there is a new user or a user needs access to an application or directory, you can simply add the user's account to the appropriate group and

the account will inherit the group's permissions. This section of the chapter starts with managing users and then progresses into managing groups.

Tools

The following tools are used to create and manage user accounts on a Linux system.

Useradd

There are a number of steps involved in adding users to a UNIX system. By default, these may include

1. Define the new user by adding a line to the ***/etc/****passwd* file for the user and create a new User Identification number (UID). The system uses UIDs to refer to the user internally.
2. Create a password for the user by adding a line to the */etc/shadow* file.
3. Define a new group for the user by adding a line to the */etc/group* file Group Identification number (GID), which is used to refer to the group in the same way the UID is used to refer to the user.
4. Create a new *home directory* for the user, set the file permissions on it and copy the default startup files to it.
5. Set up the users e-mail account.

> **Note**
> The *passwd*, *shadow*, and *group* files are discussed in the next section.

Although it is entirely possible to do each of these steps by hand, the *useradd* tool automates them. Not only does this make your life easier, it also reduces the chances of messing up the arcane file formats involved. The syntax is as follows:

```
useradd [options] username
```

Good options to know are as follows:

- `-c` or `--comment` this comment can be used to enter any text string, but it is typically used for the user's full name or a short description.
- `-b` or `--base-dir` base-directory is the directory that the user's home directory will be placed in, if you'd like it somewhere other than */home.* By default, the username is used for the home directory.

- `-m` or `--create-home` will create a home directory for the user, and copy the basic user settings files from /etc/skel, which is covered in the next section.

A number of other parameters can be set in the /etc/login.defs and /etc/default/useradd files, including the following:

- The range of UID and GID numbers
- Where the users e-mail file is stored
- Account and password time limits

userdel

Use `userdel` to delete a user. The format is straightforward:

```
userdel [options] username
```

The only two options are given below.

- `-r` or `--remove` will delete files in the user's home directory and any files it contains as well as their e-mail spool.
- `-f` or `--force` is the "nuke it from space" option – it will delete the user account even if they are currently logged in, delete the home directory even if other users may be sharing it, and potentially delete a group that matches the username even if it is used by others on the system. The `man` pages advise caution when using this option.

Note that neither of these will remove files outside of a user's home directory, so it may be necessary to use the `find` command to hunt down files that may be left orphaned.

```
find/-user username
```

The above `find` command will look up the username in the /etc/passwd file, but if you've already removed the user, you'll either have to already know the UID and use it instead of the username (the `-user` option will accept either, or you can use the `-uid` option) or use the `-nouser` option to hunt down orphaned files.

Usermod

The `usermod` command changes (modifies) a user account. The syntax is as follows:

```
usermod [options] username
```

Some of the handier options include

- To change a username, use the `-l` or `--login` option. This is helpful if your usernames include the last name or initial and a user gets married. This option doesn't rename a user's home directory, however, which ought to be done to avoid confusion. You can also use the `-c` or `--comment` option to update the comment field of the /etc/passwd file if you use that to track the user's real name.
- The `-d` or `--home` option changes the user's home directory but needs the `-m` option to move their files to the new location.
- The `-u` or `--uid` and `-g` or `--gid` options change the user's UID and default group name or number but doesn't actually change the ownership information of any existing files they may have; you have to do that yourself using `chmod`, discussed in the upcoming "tools" section.
- The `-G` or `--group group1[ ,group2,...]` option changes the groups that the user is a member of. The user is removed from any groups not listed unless the `-a` or `--append` option is also used.
- The `-L` or `--lock` and `-U` or `--unlock` options lock and unlock the account.

passwd

The `passwd` command is used to change passwords. Users can use the command themselves to change their own password, or it can be used by the system administrator to change passwords on their behalf or to reset a forgotten password by providing the username. Like `usermod`, `passwd` can also be used to lock and unlock accounts with the `-l` and `-u` options.

> **Exam Warning**
>
> The `passwd` command isn't actually listed on the CompTIA list of exam topics, but they do refer to "lock," so it is advisable to be familiar with both `usermod -L` and `passwd -l`.

Groupadd

Many of the user commands that refer to groups require that the group already exists. The `groupadd` command creates groups, and although there are a couple of options available for specifying GID or allowing shared GIDs, it is normally just used with the group name. Group names must

- Begin with a lower case letter or underscore.
- Can only contain lower case letters, underscores, or dashes.
- May end with a dollar sign.
- Can't be longer than 16 characters in total.

The syntax looks like this

```
groupadd [options] groupname
```

By default, the `groupadd` command uses the next unused GID number; the `-g` or `--gid` option lets you pick a specific number – because every company has a group that's earned the right to use GID 666.

Groupdel

The `groupdel` command is used to delete groups by removing the appropriate lines from the /etc/group and (if used) /etc/gshadow files. It won't let you remove a user's primary group; you'll have to give the `usermod --gid newgroupname username` command to form a new group. Similar to `userdel`, `groupdel` won't change the actual GID information on existing files, either. Although that doesn't cause immediate problems, if the GID gets reused, you may get unexpected file access and other security issues, so it's best to use the `find / -group groupname` or `GID` or `find / -nogroup` to track down potential problem files. The syntax for `groupdel` is simply

```
groupdel groupname
```

> **Note**
> Note that `groupdel` only works with actual group names, not group ID numbers.

Groupmod

The `groupmod` command is used to change a group's GID or groupname. The syntax is as follows:

```
groupmod [options] groupname
```

The common options are given below.

- `-g` or `--gid GID` to change the GID
- `-n` or `--new-name newgroupname` to change the group name

Remember that the most common group edit, actually changing the users in the group, is done with the `usermod` command, as explained previously.

who and whoami

Once you have a bunch of users on your system, you'll want to keep track of them. The `who` command lists the usernames of people logged into your system. Available options can show where remote users are logged in from and the number of users. The syntax is simply

```
who [options] [FILE | arg1 arg2]
```

Who works by looking in the /var/run/utmp file, which keeps track of who is logged into the system. The `FILE` option allows it to check other similar files.

> **Note**
> The `users` command gives similar, although briefer, information.

The `whoami` command simply prints your username.

w

The `w` command picks up where `who` leaves off, giving not just the user information, but also showing the following:

- What device they logged into.
- Where they logged in from (console if local or IP address if remote).
- When they logged in.
- How long their session has been idle.
- How much processor time they've used.
- What program they are running.

It includes a header that gives information similar to `top`, which shows the current time, system up time, number of users, and average load statistics (see Figure 10.1).

The syntax for `w` is as follows:

```
w [options] user
```

The options just modify which fields are viewed. Adding a specific username will limit the output to information about that single user.

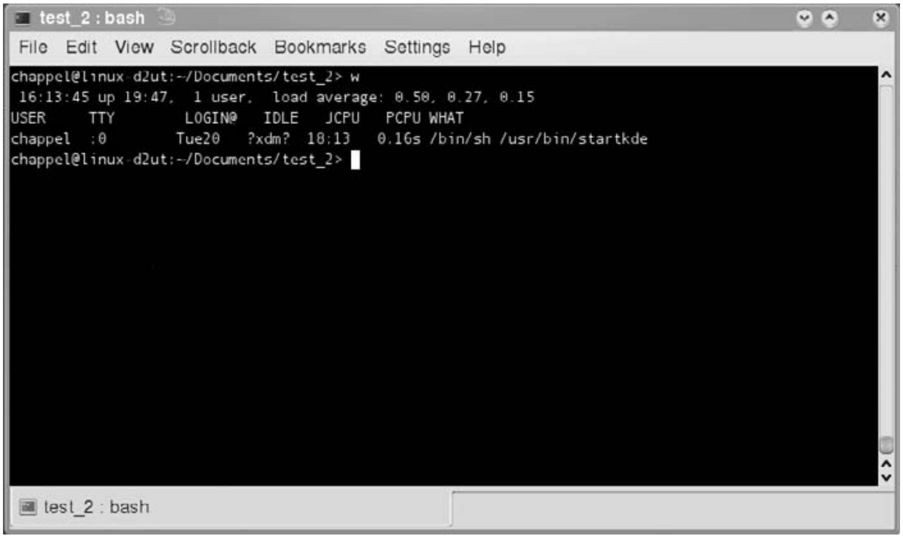


FIGURE 10.1
The `w` command.

last

The `last` command reviews the /var/log/wtmp file to show information about who has logged in (and out) since the file was created. The syntax is

```
last [options] [name] [tty]
```

Interesting options include

- `-t` <u>`YYYYMMDDHHMMSS`</u> can be used to show who was logged in at a specific time.
- Providing a <u>name</u> will give the login information for a specific user account. The system logs in with a "psuedo user" account called **reboot** each time it gets rebooted, so `last reboot` will show a list of times the system has been rebooted since the creation of the /var/log/wtmp file.

> **Note**
> Although not listed in the exam criteria, the related command `lastb` works in the exactly same way as `last`, except it searches the /var/log/btmp file and shows failed logins.

> **Note**
> A few more interesting commands are `id` and `lsof`. The `id` command has options to show information about your UID, GID, and security information. The `lsof` command

(list of open files) will show all files that are in use, who has them open and lots of other helpful information.

Files

We've touched on a number of files the Linux uses to maintain user information; now, we'll take a closer look at a couple of them.

/etc/skel

Each user account has a home directory. In addition to their documents and other files, it is used to keep their individual account settings and preferences. To help keep things standardized and give new users a good starting point for their settings there, a set of template files is copied to the new users home directory by `useradd`. These templates are kept in the /etc/skel (short for "skeleton") directory.

A typical list of files in /etc/skel looks like Figure 10.2.

The template files can be used to give users helpful defaults for their BASH shell, standardized options for a company preferred text editor, and other standardized settings. It can even be used to give company standard browser favorites or a company directory file.

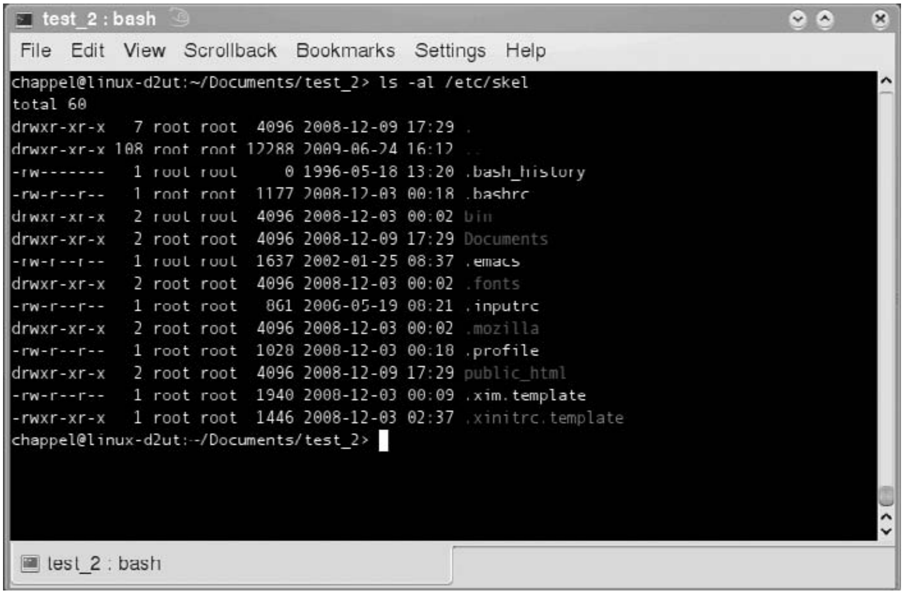


FIGURE 10.2
Files typically found in the /etc/skel directory.

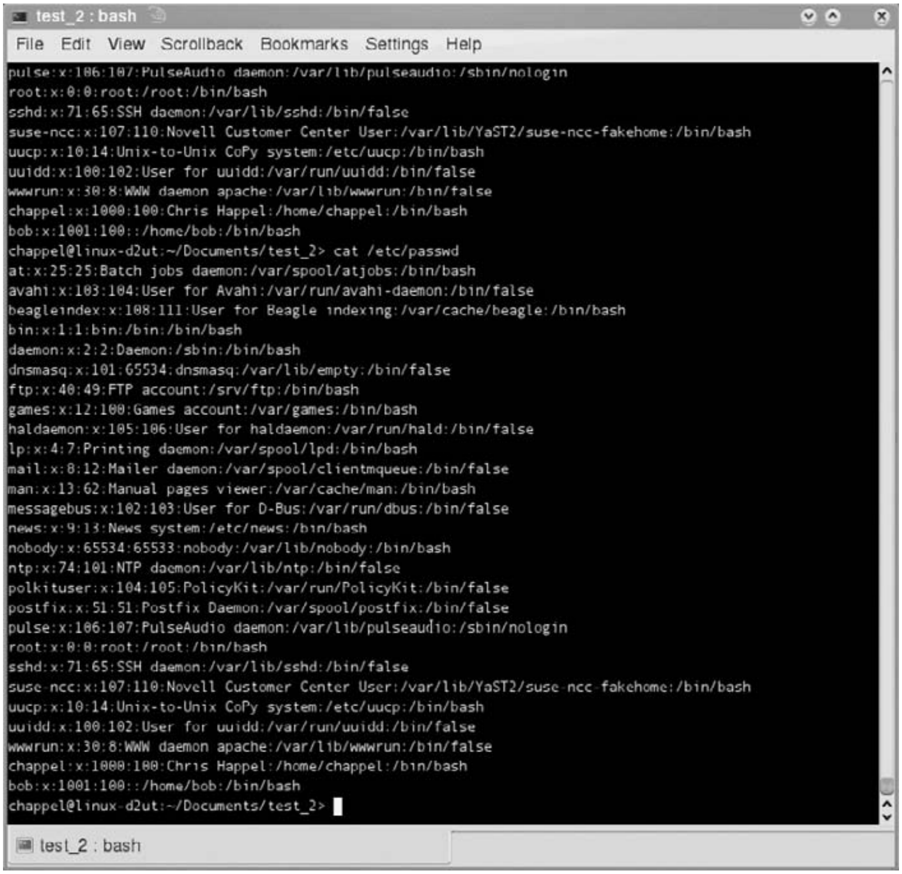
/etc/passwd

Actual user account information is kept in the /etc/passwd file. The passwd file is just a text file and can be viewed or edited like any other text file (see Figure 10.3); however, it can only be edited by a user with elevated privileges, such as "root."

Each user has its own line within the /etc/passwd file. Each line is separated into fields by colons. The fields, in order, are as follows:

- username
- password – more on this in the */etc/password* section
- UID
- GID

FIGURE 10.3
Sample contents of a /etc/passwd file.



- comment(s) – usually the user's real name, plus optional office location, phone number, or other information separated by commas
- user's home directory
- user's default shell – "/bin/false" here means that user doesn't get shell access, which is good security policy for many system accounts

The `useradd` tool is the preferred method for adding or changing user information, although simply opening the file with your favorite text editor will work, too.

> **Note**
> If you are on a busy system, use the `vipw` command to pop the /etc/passwd file open in `vi` and lock it until you are done, so multiple administrators can't make changes at the same time, which could lead to problems.

/etc/shadow

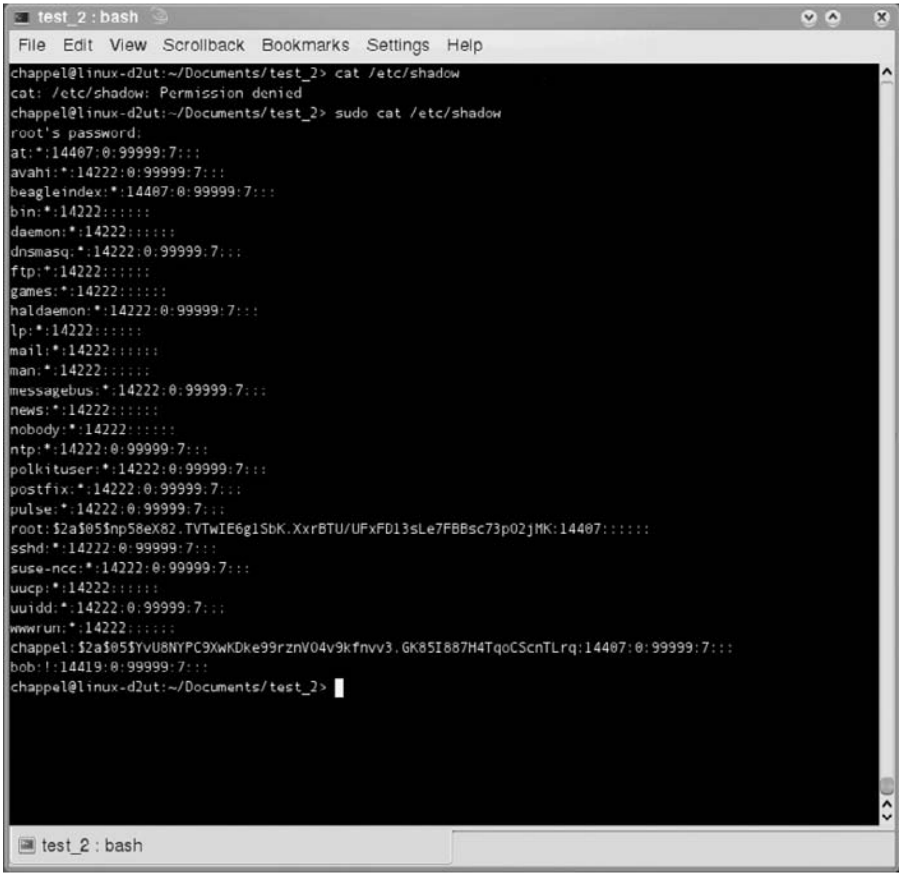
The /etc/passwd file contains lots of information that's pretty useful and is used by many common utilities to cross-reference username and UID/GID information and other things, so it is pretty handy to allow it to be read by a lot of programs. However, having easy access to every user's password, even if encrypted, is a bad thing. To solve that problem, nearly all current UNIX systems use a mechanism called a *shadow password* file, /etc/shadow. This file can only be read by system administrators and contains the actual passwords and other sensitive account information. It can also be read by processes and utilities that are setuid 0. The format is similar to /etc/passwd, one line per user (see Figure 10.4).

The colon delimited fields are as follows:

- username – matches the username from /etc/password.
- encrypted password – an * or leading ! indicates the account is locked or temporarily disabled.
- a series of encoded dates that track when the password was last changed, when it expires, and other security timing information.

While the /etc/shadow file can be edited by hand, it usually isn't for purely practical reasons. The password is encrypted by the `crypt` function, and the dates are tracked by the number of days since 1970; neither is user-friendly. Instead use the `passwd` command to change the password and `usermod` to change password dates. If you recall the password field from the /etc/passwd

FIGURE 10.4
The /etc/shadow file.



file, an *x* indicates that a password has been set and can be found in the /etc/shadow file.

> **Exam Warning**
>
> If you hand-edit the /etc/passwd file, be careful not to leave the *password* field blank. This clears the password, so anyone can connect as that username without a password. This is considered a poor security practice, even if other steps have been taken to limit that account.

/etc/group

Much like the /etc/passwd file, /etc/group contains a list of groups and related information, one group per line (see Figure 10.5).

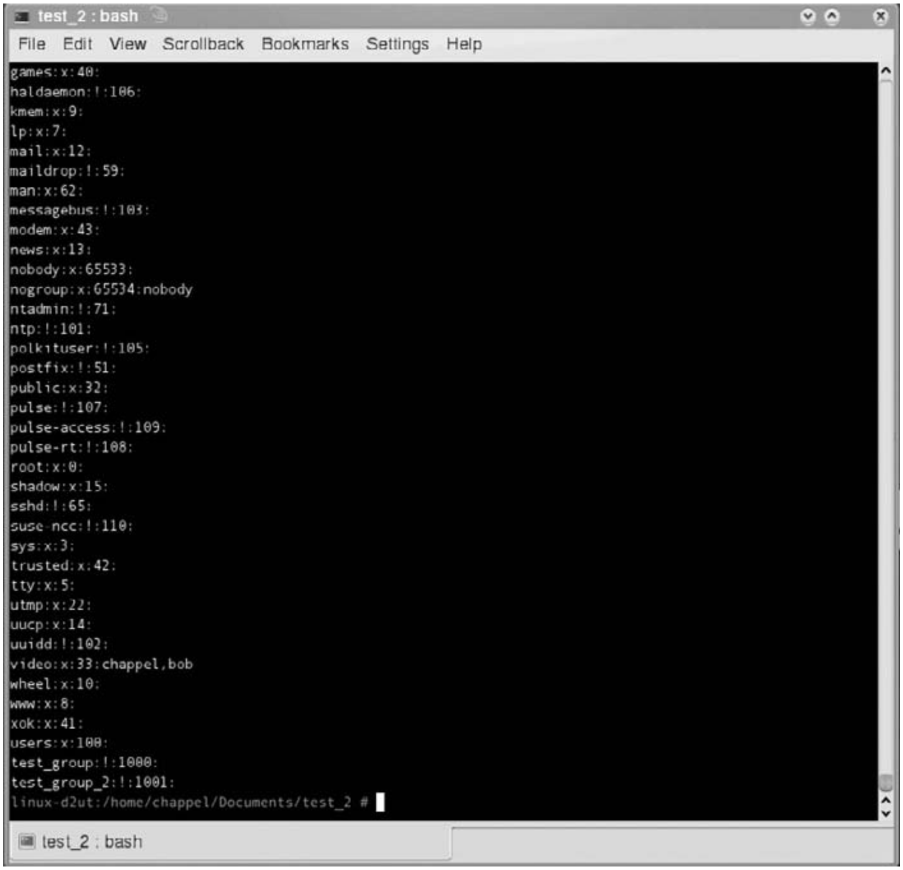


FIGURE 10.5
An example of a /etc/group file.

The fields in the /etc/group file are also separated by colons and are as follows:

- groupname
- password
- GID
- members – separated by commas

Again, there is nothing wrong with editing the file by hand, but the `groupmod` command may be easier. You'll note that there is also a field for a group password; groups use a similar mechanism as users to set a password, which can be used to delegate group management to member users.

The `gpasswd` command manages the passwords, which can be securely stored in the /etc/gshadow file.

EXERCISE 10.1: Creating and Managing Users

In this exercise, we'll use some of the user management tools. Try the following logged in as administrator at a command prompt:

1. `adduser -m tom` to create a user named tom with the next available user ID and create a home directory of /home/tom.
2. `ls /home` to confirm his new home directory is there.
3. `tail /etc/passwd` to confirm his new entry in the user list file.
4. `tail /etc/shadow` to check the secure password file. Note the "!" in the second field, indicating a password hasn't been set yet.
5. `su passwd tom` will let you set a password on tom's account.
6. `tail /etc/shadow` should now show an encrypted password on tom's account, and he'll be able to log in.

This certainly works for making a user account, but many distributions offer handy all-in-one utilities. In SuSE, you can use YAST2 to manage users and groups. ■

FILE PERMISSIONS AND OWNERSHIP

As you recall from the `ls -l` command in Chapter 6, "Using BASH," UNIX file systems have a mechanism for tracking who can do what with each file. This consists of a set of nine bits and is called the file's *mode*. The first bit represents the ability for the files owner (*User*) to *Read* the file, the second bit the User ability to *Write* the file, and the third is the ability for the **User** to *eXecute* the file (if it is a script or other executable file). The next three bits define the same rights for members of the files *Group*, and the third set of three defines what everyone else can do with the file – *Read*, *Write*, or *eXecute*. As they appear in the `ls -l`, a file with all the bits set looks like: -rwxrwxrwx (the leading dash is where the file type bit shows up, "-" for regular files). Any bit not set is shown as a dash, so -rwxr-x--- allows the user who owns the file to Read, Write, and eXecute, all members of the group associated with the file to Read and eXecute, and limits everyone (else) to no access at all.

Tools

There are a number of commands used to manage permission bits and other file attributes. As a system administrator, you'll be using them regularly, not to mention they are covered extensively on the exam, so pay particular attention.

chmod

The `chmod` command is used to change the permission bits themselves. The syntax is as follows:

```
chmod [options] MODE FILE
```

There are two particular handy options

- `-R` or `--recursive` will make changes in the entire subdirectory tree.
- `--reference = file1 file2` will set the permission bits on file2 to match file1.

There are also two very different ways to represent the mode: using numbers and using letters.

Using numbers is very quick, but a little tougher to get used to. It uses a three digit base-8 (octal) number, with each digit representing the eight binary options, the first digit for the owner, second for the group, and third for everyone else (other). All possible combinations are shown in Table 10.1 (below) from "The Linux Administration Handbook."[3]

Hence, to set file3 so that only owner and group can read and write to it, you would enter

```
chmod 660 file3
```

Table 10.1 File permission modes in Octal, Binary, and text

Octal	Binary	Permissions
0	000	---
1	001	--x
2	010	-w-
3	011	-wx
4	100	r--
5	101	r-x
6	110	rw-
7	111	rwx

Once you get used to using numbers, the option to use letters seems clunky, but it does have one advantage: when using numbers, ALL the bits get set at once, where with letters you can adjust individual parameters one at a time. The specific format has a lot of combinations of options, but is basically

```
chmod who what changes filename
```

The who can be

- u for user
- g for group
- o for everyone (think "other")
- a for all of the above

The what changes bit has two parts

- + to add a permission
- - to remove a permission
- = to set all the permission bits as shown, which is analogous to the way the octal format works

The other part specifies which bit to set

- r for read
- w for write
- x for execute

To put all that together, you list the letter(s) for the who, then the +, -, or =, then the combination of r, w, and/or x that you want, and finally, the file name you want to change. Multiple changes can be made by separating them with commas. To change the file used in the previous example to remove the ability of group members to write to it, you would use this

```
chmod g-w file3
```

To allow the user and group to execute the same file, you would use this

```
chmod ug+x file3
```

Learn by Example: Managing File Permissions

Suppose you have a file called *shared.doc* that you'd like everyone in your group to be able to read, but only you get to write to it, and everyone else can't do anything with it. In other words, you want user rights set to read and write, group rights to read,

and other rights left blank. Use `ls -l shared.doc` to check the current file mode. Now, consider for a moment that in each role (owner, group, other), you add a four to read, a two to write, and a one to execute. For the owner, you want both read (4) and write (2) for a total of six in the first position. The group bit in the next position only gets to read, so it gets a four. The mode position for "other" doesn't get any rights and stays at zero. Putting it all together, you get a mode of 640, and the numeric command to set that mode on the file is `chmod 640 shared.doc.`

Now, suppose the requirements change, and you'd like everyone in the files group to be able to edit the file and everyone else to read it, with the other permissions staying the same. That would add two to the group position and four to the position for "other," making the file mode 664. You can still use the numeric form of the command, `chmod 664 shared.doc`, but the alphabetic form of `chmod g+w,o+r shared.doc` also works. With time and practice, this will become second nature, and you'll start to recognize file modes on sight.

Exam Warning

The execute bit has to be set before a binary executable file can be run, but because a script has to be opened for the interpreter to see the commands inside, it needs both the read and execute bits set.

chown

The `chown` command is used to change a files owner and group. The syntax is as follows:

```
chown [OPTION] [username][:[groupname]] filename
```

The most useful options are the same as for `chmod`

- `-R` or `--recursive` will make changes in the entire subdirectory tree.
- `--reference = file1 file2` will set the owner and group on file2 to match file1.

If you are only changing the owner, the colon is optional. Otherwise

- `username:groupname filename` will change the files owner to username and the group to groupname.
- `username:filename` will change the files owner to username and change the group to the users primary group (note the colon after the username).

- `:groupname filename` will change the files group to groupname. This is the same as using `chgrp` (discussed next, in the chgrp section).

You can use the numeric UID or GID in place of a user name or group name.

chgrp

The `chgrp` command works the same as `chown :group`, as mentioned earlier, although it may be easier to remember. The syntax is also similar but without the colons

```
chgrp [OPTION] groupname filename
```

It also includes the same useful options of

- `-R` or `--recursive` will make changes in the entire subdirectory tree.
- `--reference = file1 file2` will set the group on file2 to match file1.

chroot

The `chroot` command is a bit different than the preceding commands. It doesn't change any of a file's attributes; instead it changes how much of the file structure a program is allowed to see. It does this by redefining the top of the directory tree (root) to wherever you specify. It is most often used to box in a program that is exposed to the public Internet, so that if it should be compromised, the attacker only has access to a minimum portion of the computer. It is often referred to as a "jail" or a "sandbox." The syntax is

```
chroot newroot [command]
```

This is normally used as a function within a script, but if the command is left off, `chroot` will give you a bash shell with the root you specified.

lsattr

The `lsattr` command is used to display a set of *attributes* that may be set on files and directories stored on common Linux drive formats (ext2 and ext3 in particular). These attributes define a number of advanced options and features that the computer uses when accessing information. The attributes include the following:

- `i` means a file *immutable*; only the root user or privileged kernel processes can make changes to it. This essentially locks a file that you don't want to be changed.

- `a` makes a file *append-only*, so it can only be added to. This is handy for log files, making it harder for an intruder to hide their tracks.
- `d` marks a file to be skipped by backups (*d* for *dump*, a basic backup program).
- `c` marks a file for compression at the kernel level.
- `s` marks a file so that it gets overwritten with zeros when deleted to enhance security.
- `A` tells the system not to use *atime* to update the access time on a file. This is occasionally used to reduce unnecessary writes to logs on a flash-memory-based file system, which have a limited number of write-cycles. Although the "limited" number is huge, updating a set of files every second burns through them unnecessarily quickly.
- `S` tells the system to immediately write any changes to the file, instead of caching them. This makes the file less likely to be impacted by a system failure or power outage that may occur between the time a file is changed and when the system gets around to writing that change. It reduces performance a bit, but may be worth the trade off for a particularly important file.
- `D` does the same as `S`, but for directories.
- `u` marks the file to allow it to be undeleted.
- `H` indicates that a file uses special block sizing to allow it to be larger than 2TB (think HUGE).

The syntax for `lsattr` is

```
lsattr [OPTIONS] [filename]
```

Useful options include

- `-d` will list directories but not their contents.
- `-R` will recursively list subdirectories.

chattr

The `chattr` command is used to change the attribute bits that were covered in the `lsattr` section. The syntax is

```
chattr [OPTIONS] [mode] filename
```

The mode is a +, -, or = to add, remove, or set exactly a list that consists of valid option letters described in the `lsattr` section earlier. An example is

```
chattr +dcs /home/bob/temp_stuff
```

This would mark bob's temp_stuff file to not be backed up, be compressed, and when it gets deleted, the space it used to occupy would be overwritten with zeros.

Exam Warning

Not all of the available options may be incorporated into a particular Linux kernel and may have unexpected results even if they are. Additional research should be done before testing any of these on a production system.

EXERCISE 10.2: Using Attributes

In this exercise, we'll test out the append-only attribute. In a command-line session with administrator privileges, type the following:

1. `touch test_file` to create a test file.
2. `lsattr test_file` to confirm no attributes are set.
3. `vi test_file, i` to insert some text, the `<esc> wq` to save the file.
4. `chattr +a test_file` to set the append-only attribute.
5. `lsattr test_file` to see the attribute bit set.
6. `vi test_file` again, `i` to add some more text, `<esc> wq` to save the file. Note that this time, it won't let you. Use `q!` to quit without saving.
7. `rm test_file` to try and delete it.
8. `ls >> test_file` to append a directory listing to the file.
9. `cat test_file` to show that you have successfully appended to the file, even though you can't edit or delete it.
10. `chattr -a test_file` to remove the append-only attribute bit.
11. `rm test_file` – can you delete it now? ■

umask

The `umask` command is used to set the default file permissions that a file gets when it is first created. Unlike the `chmod` command, `umask` uses a list of octal values to indicate what rights to *remove*. A typical umask is 0022 (the first bit

is for special permissions, discussed next). The two mean new files will have the write privilege removed for members of the files *group* and *other*.

You can view your umask by simply typing the `umask` command or change it by using

```
umask newmask
```

You can also use `umask` with the same letter syntax as `chmod` by using the `-S` parameter (for Symbolic). It will also accept letters to set the mask. Using letters tells the system which bits to set, as opposed to which bits NOT to set for the number representation, which may be less confusing to use.

To make a change permanent, you can add the command on your shell startup script, so it gets run every time you start a shell.

EXERCISE 10.3: Using `umask`

In this exercise, you'll see a quick use of `umask`.

1. `mkdir` test to create a test subdirectory.
2. `umask` to show the current mask – probably 0022.
3. `touch test_file1` to create a file.
4. `ls -l test_file1` to see the file mode.
5. `umask 0000` to change the mask.
6. `touch test_file2` to create a new file, using the changed mask.
7. `ls -l` to compare the mode of your two files. ■

Special Permissions

In addition to the normal permission bits, there are additional *special permissions* that are represented by a fourth octal number, put in front of the normal three. Linux figures out what you want automatically – if you use four numbers, it understands the first one is for special permissions and if there are only three numbers, it sticks with the regular read, write, and execute permissions.

Similar to the normal permissions, an octal value of 4 is the *setuid*, 2 is the *setgid*, and 1 sets the *sticky bit*. They are set using the same `chmod` command as the normal bits, either using a four-digital octal value or the same method using letters, but with the following additional letters:

- `X` sets the user or group ID
- `s` restricted deletion flag
- `t` sticky bit

The special permissions have different meanings depending on whether they are set on a file or a subdirectory, as described below in the *setuid* section.

setuid and setgid

When the *setuid* or *setgid* bit is used on a directory, it will cause files created within the directory to have the UID or GID of the directory, instead of the user who created the file. This is convenient for shared directories, so everyone who is a member of the directory group can easily share files.

When the *setuid* or *setgid* bits are set on an executable file, the program runs with the privileges associated with the program owner or group membership of the program and not the user who is running it. This allows users to run programs that can get to things the user wouldn't be able to directly.

As an example, consider the `passwd` command, which allows users to change their passwords. This command needs to read and write to the /etc/shadow file where users' passwords are stored, a file that users don't have access to. Using the setuid bit, users are able to run that specific command with the necessary administrator rights to get the job done. Try `ls -al /usr/bin/passwd` to see the setuid bit.

sticky bit

The third special permissions bit is called the *sticky bit*. When this is set on a directory, files within that directory can only be renamed or deleted by their owner, the directories owner, or the system administrator. Without the sticky bit, any user with write and execute privileges in the directory could delete files. This is another handy feature to have in shared directories, which limits what users can do with shared files owned by other users.

SELINUX BASICS

As mentioned in the beginning of the chapter, Linux is considered a secure operating system. There is always room for improvement, though, and the goal of Security Enhanced (SE) Linux is to greatly reduce the possibility of security problems on Linux-based systems.

One of the problems with most current computer systems – including Linux – is that they are dependent on the software provider to recognize and then repair potential security problems, then the system administrator has to find and apply any fixes. Computers can be configured to automatically find and apply updates, but that can introduce additional problems. The real risk is that someone figures out how to exploit a vulnerability in your system before you get a chance to patch it or even before the software provider realizes there is a problem and can fix it. These security problems

are commonly called *0-day* exploits, because the good guys may get zero days prior notice to fix a problem before there is already someone using it to wreak havoc.

SELinux reduces the risk of these problems by implementing a system that limits what programs can do. Ideally, no user or program should have access to anything more than it needs to do its job – this is a standard security paradigm called *least privilege*.

In his book "SELinux," Bill McCarty points out that normal Linux uses a system called *discretionary access controls* (DAC) meaning the security level depends on the user. If user Bob runs a program with his login, that program typically has access to everything that Bob has access to. If someone manages to compromise Bob's program, now THEY have access to it, too.

Bill further notes that SELinux uses the more secure *mandatory access controls* (MAC). This is a mechanism that enforces least privilege access by program. Now, if Bob's program is compromised, it only affects that program. That's still not good, but it's a lot better than having the entire system crack open. It's the difference between a burglar breaking into a single safe deposit box instead of an entire bank.

Running Modes, Enabled, Disabled, Permissive

Current versions of the Linux kernel have the necessary components to support SELinux, it just needs to be enabled, which varies by distribution. Once enabled, SELinux defines categories of Subjects, Objects, Actions, and sets up rules that define how these categories are allowed to interact.[4]

To make it easier to set things up, SELinux provides three *running modes*:

- *enabled* – SELinux is up and running, any forbidden actions are blocked.
- *disabled* – SELinux is there, but not turned on.
- *permissive* – SELinux is up and running, with the rules in place, but when something forbidden is attempted, SELinux allows it. A log of rule violations is kept, which can be used to adjust the rules to make sure valid activities aren't obstructed.

When first setting up SELinux, the permissive mode helps identify conflicts with the security settings. Once everything is running smoothly, it can be set to *enabled*.

IMPLEMENTING PRIVILEGE ESCALATION

As a system administrator, you need to be careful what you do; even a small typo can cause devastation. Consider the difference between

`rm -r / tmp/test` and `rm -r /tmp/test`. The difference of a single errant space, in this case, will tell the system to try and delete every file on your machine. To protect your system as well as to use the idea of "least privilege" discussed in the previous section, it's advisable to use your system with normal user privileges whenever possible. For those times, when extra authority is called for, you can use the power of *Privilege Escalation*.

sudo

The standard way of getting extra administrative power from a terminal window is to use `sudo` (think "**do** as **s**uper **u**ser"). It gets put in front of whatever command you want to run, and after you press the **enter** key, it will ask for your regular user password and run the command as administrator. The syntax is

```
sudo [OPTIONS] command
```

Some of the more interesting options include

- `-u` will let you run a command as a user other than the system administrator. This is handy for troubleshooting problems that involve user privileges.
- `-e` lets you enter a file to be edited instead of a command.

As a convenience, once you authenticate with `sudo`, it will remember the password for a few minutes, so additional uses of `sudo` don't ask for a password.

su

Even with the several minute password grace period it can become tiresome to keep typing `sudo`, so if you need to do a bunch of stuff, it is often easier to start an entire new command shell with administrative privileges with the `su` command. Unlike the `sudo` command, `su` uses the administrator password. The syntax is

```
su [OPTIONS] [username]
```

As with `sudo`, `su` can be used to open a shell as any other user, too.

> **Note**
> Note that `su` opens a whole new shell, as a different user. Because of this there is a completely separate history file, so you can't use the "up" arrow to get to the command that you just got denied from running because you weren't root. Typing `exit` or **<ctrl>d** will put you back to your normal user shell.

/etc/sudoers

You wouldn't want to give just anyone the awesome power of *super user*. Only users who are listed in the /etc/sudoers file can use privilege escalation. It also provides a number of options.

Although the /etc/sudoers file is just another text file, it is recommended that you use `visudo` as root to make any changes. This opens the sudoers file in `vi` and protects it from simultaneous edits, similar to `vipw`. It also does a sanity check on the file before saving to ensure syntax is correct.

The options are typically described in commented sections within the file (see Figure 10.6) and include limits to what files each user can run, as well as the very bad option of allowing users to assume administrator privileges without a password.

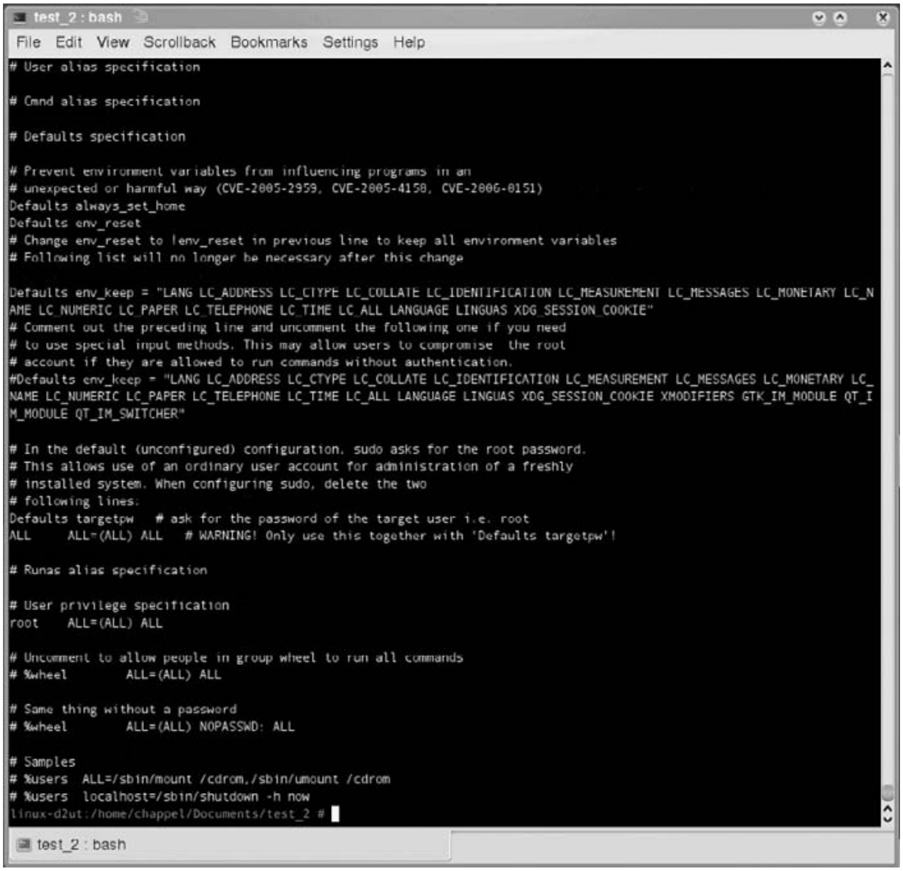


FIGURE 10.6
A typical example of /etc/sudoers.

EXERCISE 10.4: Assuming Administrator Rights

In this exercise, we'll escalate your user privileges to administrator level using `su`. In a terminal window running under a normal user account, type the following:

1. `cat /etc/shadow` – as a normal user, this is a restricted file.
2. `su`.
3. enter the administrator (root) password.
4. `cat /etc/shadow` – now with your administrative privileges you should be able to see the file. Tread carefully, and type `exit` when you are ready to return to your normal user level. ■

SECURITY APPLICATIONS AND UTILITIES

In addition to the start management utilities for keeping a Linux system secure, there are a number of additional software packages that can be used to keep a close eye on your system and network. Each of the following tools are open source and available online (except Nessus, see below) through a standard package manager. They are all very complex and feature rich, and each is the topic of multiple books. For the exam, it is important to know what each tool is used for, although additional reading about each will be greatly beneficial to any professional system or network administrator.

Exam Warning

These tools are very useful for protecting your systems, but like any power tool, they can be misused easily. Be sure you have full authorization – preferably in writing – before using them outside of your own test environment. The tools that send out test packets are capable of sending information that can cripple or reboot some systems, which is a great way to test whether a system is vulnerable but could lead to a lot of problems. Please use these tools with care.

nmap

`Nmap` is an open-source network scanning tool. It is invaluable for seeing what is attached to your network and can test any attached system for services or open network ports and works great for finding unauthorized network devices and services and testing firewalls. It can use known differences in replies to different packet types to guess what operating system is running

on each device and can scan both User Datagram Protocol (UDP) and Transmission Control Protocol (TCP) network ports. It can be installed using your distributions package management system, or downloaded directly from http://nmap.org.

The advanced features and options available with `nmap` are well beyond the scope of the exam, but the basics are as follows:

```
nmap [Scan Type] [OPTIONS] target
```

A typical use of `nmap` is

```
nmap -sP 192.168.1.0/24
```

This will quickly use the `ping` ICMP utility to check for active hosts on the entire 192.168.1.1 – 192.168.1.254 subnet.

Use `man nmap` or check the extensive documentation on the `nmap` Web site for additional details.

EXERCISE 10.5: Using `nmap`

In this exercise, we'll do a quick network scan using `nmap`. If your system doesn't have `nmap` installed already, you should be able to easily add it using the standard package manager. If not, you can download it from http://nmap.org. In a terminal window, type the following:

1. `nmap 127.0.0.1` is an easy place to start; it will scan the machine you are on.
2. `netstat -l` will also show ports on your local machine that are in a "listening" state, for comparison.
3. `netstat -r` will show the subnet you are attached to; use that address for the network in the next step. If you are using a typical home wireless route, there is a good chance your network will be 192.168.1.0. I'll use that as an example.
4. `nmap 192.168.1.0/24` to scan the local network or enter a specific host address without the "/24" subnet mask bits to scan a single machine. ■

Wireshark

Wireshark is a graphical network traffic analyzer built on the text-based `tcpdump` utility. Both are open sourced and available through standard package managers.

Wireshark can be used to monitor all network traffic coming in and out of an interface on your computer. Note, however, that to optimize performance,

most modern network equipment limits the traffic that your individual workstation has to see. Most managed switches have an option to allow all traffic to be copied to a given port for monitoring and troubleshooting; enable this option on your switch to see the full power of *Wireshark*.

The main *Wireshark* window is split into three panes. The top section shows one line for each packet; the middle section shows a textual description of each portion of a specific selected packet, and the bottom section shows the raw hex information for the same packet.

Watching all the traffic can be fascinating, but overwhelming. *Wireshark* provides a complex way of filtering out traffic you don't want to see based on specific hosts or protocols.

Although *Wireshark* is clearly a fantastic network troubleshooting tool, it is also very useful for fixing network applications, such as client-server database problems, remote access authentication issues, and printing errors. Frequently, the client software hides error details from the user, and a lower level view is needed to isolate the real issue.

As with the other tools, use *Wireshark* with care, as it is possible to view other user's passwords or private information; keep in mind that many applications, including e-mail, don't use encryption.

More information about *Wireshark* can be found at www.wireshark.org.

Note
Although the command-line-based `tcpdump` isn't as user friendly as *Wireshark*, it has a distinct advantage if you need to check traffic on a remote system using *ssh*. It offers identical filtering and capturing features and can be just the thing to quickly confirm whether your network packets are in fact reaching a remote system.

Nessus

Nessus is a tool for testing networked systems for security vulnerabilities, which are defined as "any programming error or misconfiguration that could allow an intruder to gain unauthorized access."[5] Previous versions were released under the GNU Public License (GPL) open source license; any changes since release 3 Nessus have been kept the proprietary property of Tenable Network Security, although they continue to make the *Nessus* tool available for no charge.

Nessus can be installed and used by mere mortals, but it has the capability to scale by splitting up the part that does the scanning from the part that analyses the results, so you can install dedicated machines at remote sites and then centrally review the results.

FIGURE 10.7
A Nessus security scan.

The actual scan works by using a list of currently known security problems, which is kept updated by means of downloaded *plugins*. During the scan, the *Nessus* software checks network hosts for matches with the list of problems from the plugin and reports back the results. An example of the result of a *Nessus* scan can be seen in Figure 10.7.

This is clearly a very brief explanation of what *Nessus* does; for a more complete guide, check out Russ Rogers' *Nessus Network Auditing, 2e*, ISBN: 978-1-59749-208-9, Syngress, or visit www.nessus.org

Snort

Snort is a Network Intrusion Detection System (NIDS) and is similar to *Wireshark*; it monitors traffic on a network interface. The big difference is that *Wireshark* is meant for you to watch traffic yourself; *Snort* watches the traffic for you. *Snort* lets you set a list of traffic you might find interesting or you can download lists of known signatures of traffic you probably want to know about (like viruses and other nasty stuff). If an interesting packet wanders by, *Snort* can send out an alert so you can then investigate further.

Snort can be considered a complementary program to *Nessus*. Where *Nessus* sends packets out checking for possible vulnerabilities, *Snort* quietly listens for someone trying to abuse them.

For further information, refer to *Snort Intrusion Detection and Prevention Toolkit*, ISBN: 978-1-59749-099-3, Syngress or to check, go to www.snort.org.

Tripwire

Tripwire is a Host-based Network Intrusion System and is the only non-network utility in this list. When first installed, it takes a digital "thumbprint" of key files (more details are given in the next section) and occasionally checks to make sure they haven't been tampered with. *Tripwire* is available as an open-source project or can be purchased as a product from www.tripwire.com.

CHECKSUM AND FILE VERIFICATION UTILITIES

One particularly complex security problem involves being able to trust that files you rely on don't have any errors and haven't been tampered with. Checking for errors isn't terribly difficult and happens to every TCP packet as it passes through every router; the router "adds up" the bits, and if they match the total that the sending device claims it did, everything is assumed to be OK. If not, the packet is thrown out and a new one is requested. At the networking level, the calculations are fast and easy, and although it's possible that a random error could occur that makes the bits still calculate to the same amount, it's not very likely.

When it comes to security, though, it requires protection against possible malicious intent, not just random errors, so the calculations involved are much more complex. A number of methods are currently used to certify that files haven't been tampered with; three are discussed below.

md5sum

The `md5sum` uses the MD5 (message digest algorithm number 5) to calculate a checksum for a downloaded file, which is then compared with a checksum supplied by the files creator. The syntax is

```
md5sum [OPTION] [file]
```

It is typically used without any options and spits out the 128 bit hexadecimal checksum. You can then compare your locally computed number with a provided, known good checksum to ensure you have a good copy. Alternately, you can use the `-c` or `--check` option to feed it a list of checksums and filenames, and it will confirm if everything is OK.

Unfortunately, recently, it has been demonstrated that the *md5* algorithm has problems,[6] so it doesn't have the same guarantee it once had.

For more information about the *md5* algorithm (including the code) see rfc1321.

EXERCISE 10.6: Comparing File Checksums

In this exercise, we'll use `md5sum` to compare checksums of a file. In a terminal window, type the following:

1. `md5sum /etc/hosts` to generate an md5 hash of your hosts file.
2. `cp /etc/hosts` to put a copy of the hosts file in your current directory.
3. `md5sum hosts` to generate a hash of the copied version of the file – they should match.
4. `md5sum hosts > hosts.md5` to create a file that has both the hash and the file name.
5. `md5sum -c hosts.md5` will look for the file name in the file in the current directory and compare the hash in the file with one it recomputes on the file it finds, and will return "ok" if they match. Frequently, programs will include both the hash value for a manual comparison and a file to do an automatic check. Note that the file can have many lines of hashes and files; it's an easy way to compare a lot of them in one go. ■

sha1sum

The `sha1sum` command is the same as `md5sum`, but uses a different algorithm and a longer checksum value (160 bits instead of 128). Other than that it is used in the same way

```
sha1sum [OPTION] [file]
```

The `sha1sum` command implements SHA-1, one of a family of Secure Hash algorithms, as defined by the NIST FIPS-180-2 standard. Linux also supports variations of the SHA-2 algorithm (sha224sum, sha256sum, sha384sum, and sha512sum).

So far, the SHA family of algorithms is still secure, but there are indications that it may be breakable.[7]

gpg

Another option for protecting files is `gpg` – *the Gnu Privacy Guard*. It uses an open implementation of pretty good privacy (PGP) to encrypt and/or sign files using public/private key pairs. Although it is normally used for signing and encrypting e-mail to make sure it hasn't been read or changed, it can also be used to sign files. It works like this

- The sender of a file creates a public/private key pair, maintaining the private key secret and sharing the public key with anyone who will need to decrypt or verify information from him. The key generation process only has to be done once. Sharing the key can be done in-person or using a key-escrow service.
- The recipient of the file obtains the senders public key and imports it into his system.
- When the recipient gets a file, he uses the sender's stored public key to decrypt or just authenticate the file.

There are a bewildering number of options for the `gpg` command, but to do a simple file verification use

```
gpg –verify filename
```

For further information, review the man pages for `gpg`.

IMPLEMENTING REMOTE ACCESS

You've dealt with the risks of having everything networked; now, we'll review how to securely leverage the benefits.

Originally, UNIX systems were created in an environment where people trusted each other. This has obviously changed. The original utilities used to communicate between systems, like `telnet`, generally sent information without bothering to encrypt it in any way, even passwords.

SSH

While `telnet` is still available on modern Linux systems, the preferred method for accessing a command shell on a remote system is with SSH, the Secure Shell. It is nearly as easy to use for basic shell access, is very secure, and offers several very cool extra features. There are two parts to `ssh`: the `ssh` client you use to connect to a remote system and the `sshd` "daemon" that runs on the remote side.

The syntax for ssh are as follows:

```
ssh [OPTIONS] [username@]hostname [command]
```

The list of options is truly bewildering, but for basic connectivity, all that is required to connect to server1 is to type

```
ssh server1
```

This abbreviated format assumes you want to use the same username on the remote system as you are using on the local one. The passwords don't have to match. To login to the remote system with a different username, you can either use the `-l username` option or type the username in front of the hostname, with an @ between them, like this

```
ssh bob@server1
```

The first time ssh is used to connect to a remote host, the new hosts signature is shown, and ssh asks if you'd like to add the signature to your list of known hosts.

Secure tunnels

One of the more interesting capabilities of `ssh` is to catch traffic going to a local TCP port, pass it through its own encrypted connection, and hand it off to a TCP port on the remote side. With this feature, it is possible to protect normally unencrypted network traffic. This is called forwarding or tunneling.

To create a tunnel, you need to know the TCP port of the service on the remote server that you want to tunnel to and pick a random unused TCP port on your local service. The syntax looks like this

```
ssh -Llocal_tcp_port:localhost:remote_tcp_port remote_host_name
```

An example of this is shown in the book *Next Generation SSH2 Implementation: Securing Data in Motion*, ISBN: 978-1-59749-283-6, Syngress,[8] to forward SMTP traffic between your local e-mail client and your Simple Mail Transfer Protocol (SMTP) mail server http://my_mail_server.com using a randomly chosen local TCP port of 4444, and for the standard SMTP TCP port of 25, you would do this

```
ssh -L4444:localhost:25 my_mail_server.com
```

The final step is to tell your e-mail program that SMTP is now available on port 4444 at localhost (configuration varies depending on your choice of e-mail program). As long as your ssh session is logged in, you have encrypted e-mail between you and your server.

SFTP

Another handy feature is the ability to transfer files through `ssh` connections. Like `telnet`, the old standby of file transfer, file transport protocol (FTP), doesn't use any encryption, so using `ssh` represents a big improvement to security. Rather than using the actual `ssh` command, sending files can be done using `sftp`, which can be used interactively. Once you log in to the remote system, you can then use `sftp` commands to send and receive files. The syntax to get connected in this way is the same as regular `ssh`

```
sftp [username@]hostname
```

Once connected, you can use a number of commands that are similar to both standard BASH commands and traditional FTP commands, including

- `cd path` – to change the remote directory.
- `df -h` – to show information about the remote file system, including the remaining space (in "human readable" format).
- `lcd path` – to change the local directory.
- `ls` – to view the contents of the remote directory.
- `lls` – to view the path of the local directory.
- `lpwd` – to see the local working directory.
- `put` – to copy a file from the local to the remote machine.
- `get` – to copy a file from the remote machine to the local.
- `bye` or `quit` – to exit `sftp`.

Note that *sftp* can also be used to download files noninteractively in "batch mode" by giving it the file information all at once, like this

```
sftp [username@]hostname[:filename]
```

Unless an option automatic authentication method has been set up, the batch mode will still require a password to be entered manually.

EXERCISE 10.7: Using `sftp`

In this exercise, we'll test out copying files using `sftp`. You'll need either another system running *ssh* or you can run it on your local machine. It may be necessary to modify any firewall you are running and start the `ssh` service on your local machine. On SuSE, you can click your way through YAST or type in `sudo /etc/init.d/sshd start` to start the service. Then, type the following in a terminal window:

1. `mkdir sftp_test` to make a test directory.
2. `ls -al /etc > test_file` to create a test file.
3. `sftp localhost` – to connect to your own machine, as the user you are currently logged in as. Enter "yes" to add the RSA "fingerprint" and log in using your password.
4. ? will show a list of commands. They should be familiar – the commands are similar to what you use in a regular terminal session. Note that because you are logged into your own machine, both the "local" and "remote" are the same file system. Use `lpwd` to see your "local" directory – which will be the directory you were in when you started `sftp` – and `pwd` to see the remote directory, which will be your home directory for the user you logged into `sftp` with.
5. `lcd sftp_test` to change directories to your "local" test directory.
6. `put test_file` to copy your test file up to the "remote" end.
7. `ls` to show that your file was copied.
8. `exit` to leave `sftp`.
9. `ls` to check whether your test_file was copied to your home directory.

X11 forwarding

A third nifty feature of `ssh` that is sure to please all of you that are getting tired of using the command line is *X11 forwarding*. If you'll recall from the section on X11 in Chapter 8, "Installing, Configuring as a Workstation," the Linux graphic interface is actually made of two parts; counter-intuitively called the client, which runs on the back end, and the server, which draws all the graphics you see. *X11 forwarding* is essentially the same as the TCP port forwarding discussed earlier, but it is much easier to use. It allows you to connect your local X11 server to the remote X11 client, such that

a program running on the far end draws a graphic interface on your local screen, using just an `ssh` session. To test it, **log** into a remote host with `ssh` or `ssh -X` if X11 forwarding is not enabled – the default behavior on many distributions – and execute a program that has a graphic interface; `xeyes` is a good program to use for testing. You may want to add an & at the end of the command to run it in the background, so it doesn't tie up your session.

Keygen

The `ssh-keygen` command is used to create a public/private key pair, just like `gpg` uses (as discussed previously). Once the public key is placed on the remote host, `ssh` uses the keys to authenticate your login, and passwords are no longer required. The steps are as follows:

1. use `ssh-keygen` to create a key pair.
2. Copy the public key from your local user home directory .ssh/id_rsa.pub to the remote user home directory .ssh/authorized_keys.
3. `ssh` checks for matching keys when logging in, and if they are found, it doesn't ask for a password.

Once the keys are in place, ssh can be configured to require public/private keys to log in remotely, which makes for a very secure system, as long as your keys remain safe and don't get lost.

VNC

Virtual Network Computing (VNC) is a graphical remote desktop application that can be used as an alternative to `ssh`. It is also open source, although there are for-pay options as well. Like `ssh`, there is a client (viewer) and server component. One of the handy features of VNC is that it works on lots of different systems, including Windows, Apple OSX, and Linux as both a client or a server. The client can also be used as a Java Web browser plug-in, and there are clients available for several smart phones, so you can remotely access your computer on your cell phone.

UNIX-based versions of VNC work with X11 and support the option of either sharing an existing X11 desktop or running independently, so multiple users can each have their own remote session. The default VNC connections run on TCP ports 5900–5903 for sessions numbered 0–3, although this can be changed in the VNC configuration. Standard VNC has rather weak security, so it is generally recommended to use a VPN or tunnel it through `ssh` if you are using it over the open Internet.

AUTHENTICATION METHODS

Authentication is the process the computer uses to determine you are supposed to be given access when you type in your username and password. It comes in two basic flavors

- Local authentication is limited to a single computer. It knows who you are, but no other computers do. It is easy to set up and administer on a few computers, but scales poorly.
- Centralized authentication allows user information and other settings to be gathered into a single repository and then accessed from trusted computers. Centralized systems can be much more complicated to configure, but make it much easier to administer large networks of computers.

PAM

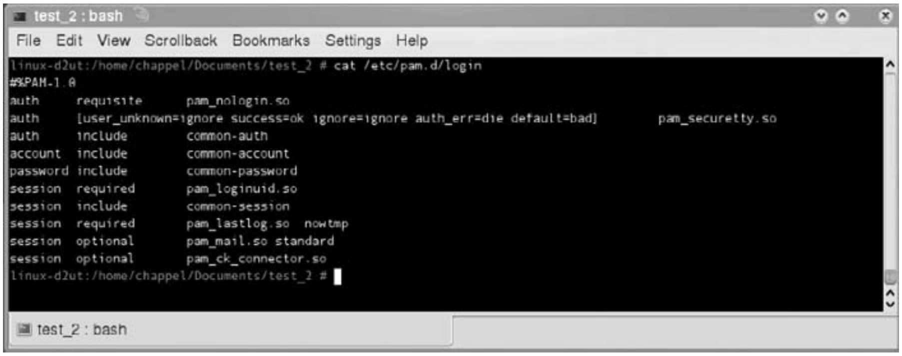
The Pluggable Authentication Modules (PAM) work to coordinate authentication requests from Linux programs. The modular design means that implementing some new authentication technology (like a fingerprint scanner) or policy (like mandatory password complexity) is as easy as plugging in the appropriate modules and telling the system to use them by updating the appropriate configuration file. Each authentication program has its own PAM configuration file. Configuration files are stored in the /etc/pam.d directory.

The Linux-PAM System Administrators' Guide[9] outlines four types of management tasks that PAM can take care of

- **Authentication management** It covers verifying users who they claim to be.
- **Account management** It allows updates to user account information, confirms account is still valid, and tracks password expiration dates.
- **Session management** It handles tasks related to the start and close of a user session, such as mounting a home directory or updating an audit log.
- **Password management** It is used to update password or other authentication mechanism.

Each PAM configuration file lists types of management tasks, which module(s) to check while performing the task, and a control that tells whether passing the module test is, among other checks, mandatory, or optional (see Figure 10.8).

FIGURE 10.8
An example of the PAM login file.



The actual PAM modules are stored in /lib/security/. Lots of additional modules and documentation can be found at www.kernel.org/pub/linux/libs/pam/.

EXERCISE 10.8: Adjusting Minimum Password Length with PAM

In this exercise, we'll enforce a minimum password length of 10 characters using a PAM policy in SuSE. With administrator privileges in a terminal window, type the following:

1. `cd /etc/pam.d` to change to the directory where the PAM configuration files are.
2. `cat common-password` to check out the default password policy.
3. `passwd` to change the root password – choose something short; five characters is the default minimum.
4. `pam-config - a --pwcheck-minlen=10.`
5. `cat common-password` to see how the configuration file has changed. It's possible to just edit the file directly, but the `pam-config` utility will overwrite any edits.
6. `passwd` to change the root password again. Note that now you'll have to pick a longer password.

Linux distributions have subtle differences in how they handle PAM configurations; it may be necessary to do additional research on other systems. In addition, note that this change can be made in the YAST GUI system management tool. ■

LDAP

LDAP is the Lightweight Directory Access Protocol. A proper understanding of LDAP requires a few definitions and a little history. A *directory* is defined by the OpenLDAP project as "a specialized database specifically designed for searching and browsing, in addition to supporting basic lookup and update functions."[10] They point out that a normal *database* is usually optimized for high volume, high speed, and often complex transactions, where a directory spends will get a lot of searches and relatively few – and pretty basic – updates and changes. Think about how often a user directory would be queried for a username and password or e-mail address compared with how frequently a user is added or a password is changed versus how often an inventory database needs to add or subtract items.

Historically, the OSI defined x.500 directories and a Directory Access Protocol (DAP) to query them. Over time, the simpler LDAP access protocol developed to take advantage of a basic subset of features from the original DAP and eventually included the actual directory. Now, the term LDAP is generally used in reference to the entire directory and not just the access protocol.

Current LDAP systems on Linux typically use the stand-alone LDAP daemon, `slapd`. This provides the back-end server functionality to store user information. Individual workstations then access the *slapd* directory information as needed: usually via a PAM plug-in for authentication or maybe through an e-mail client to look up e-mail addresses.

What does this directory information look like? Again referring to the OpenLDAP administrators' guide gives a good description:

> *"The LDAP information model is based on entries. An entry is a collection of attributes that has a globally unique Distinguished Name (DN). The DN is used to refer to the entry unambiguously. Each of the entry's attributes has a type and one or more values. The types are typically mnemonic strings, like "cn" for common name, or "mail" for e-mail address. The syntax of values depends on the attribute type. For example, a cn attribute might contain the value Babs Jensen. A mail attribute might contain the value "babs@example.com." A jpegPhoto attribute would contain a photograph in the JPEG (binary) format."*[11]

This information is organized in a hierarchy, usually with the country or organization information at the top, working its way down through departments or states finally down to the individual user or other individual piece of information, such as a workstation or server.

NIS

Network Information System (NIS) is another option for storing centralized user and other configuration files. It was originally called the "yellow pages," so many NIS commands and files start with "yp."

NIS is comprised of one or more servers that contain a database of configuration files that are to be shared across the network and clients that are configured to access them. By sharing /etc/passwd, /etc/group, and /etc/shadow files, a user's login account information will work on any client in the system.[12]

RADIUS

Remote Authentication Dial In User Service (RADIUS), as the name indicates, was originally developed for dial-in access via modems. The RADIUS client software runs on the device to which the user is attempting to authenticate. The client forwards the user information to a RADIUS server, which grants or denies the request for access. The actual password is never sent over the link.[13]

Two-Factor Authentication

Two-factor authentication refers to a requirement to provide two things when you log in. These are usually "something you know" – a memorized password – and "something you have" – an access token of some sort. Two-factor authentication makes it much harder to break into someone else's account because just guessing their password is no longer good enough.

SUMMARY OF EXAM OBJECTIVES

While "security" is frequently associated with "restrictions," it's helpful to consider it in terms of "allowances." To summarize, remember that the `adduser` command creates accounts to allow access to the system, `chmod` modifies what users can do with files, and `groupadd` allows users to share information amongst themselves. The `su` and `sudo` commands extend user privileges to allow administrative tasks, and `ssh` and `vnc` allow users access to other systems. User account information can be shared between systems with `ldap` or `nis`, and authentication requirements can be customized using on a Linux system with *pam*.

There are number of applications for observing the Linux system environment, including

- *Nessus* and `nmap` for scanning networks.
- *Wireshark* for capturing packets on a network interface.

- *Snort* for watching a network for suspicious traffic.
- *Tripwire* to watch for suspicious file updates.

SELinux is used to limit what an application can do using *mandatory access controls*, and `gpg` can be used to encrypt information or to guarantee it hasn't been altered.

SELF TEST

1. HR calls to tell you Susie Smith got married and needs her Linux user account changed from ssmith to sjenkins. What command will change Susie's user account and change her home directory?

A. `useradd -c sjenkins smith`

B. `usermod -c sjenkins -d /home/sjenkins -m ssmith`

C. `umod sjenkins ssmith`

D. `uname sjenkins ssmith`

2. Bob just came back from vacation, where he had such a good time forgetting about work that now he can't remember the password to his Linux account. Which of the following commands could you use to reset it for him if you are logged in as root?

A. `sudo -u bob passwd`

B. `passwd bob`

C. `usermod -p <new_password> bob`

D. `password bob`

3. Your boss wants you to find out how many devices are on your subnet right now because he is thinking of adding another 20 machines and doesn't want to run out of addresses. What's a good way you could check?

A. Use `wireshark` to monitor network traffic

B. Use `snort` to scan the network

C. Use `tripwire` to scan the network

D. Use `nmap` to scan the network

4. You just finished downloading the latest version of your favorite program and want to make sure that it downloaded correctly by comparing it with the hash file posted on the download site. What program would you use to make sure the file is exactly what it should be?

A. `filecheck my_download`

B. `md5sum -c my_download.md5`

- **C.** `diff my_download.md5`
- **D.** `gpg my_download`

5. You are going to start working from home occasionally and need full access to your office Linux workstation as well as a windows PC via your VPN connection. What remote access software should you consider?

- **A.** ssh
- **B.** X11
- **C.** VNC
- **D.** PCAnywhere

6. Your company is starting a new project, and you need to create a shared directory for the project team members to share their documents. You want all the new documents created in the directory to be automatically set to allow their owner and other group members to read and write them and all other system users to read only. What commands could you use?

- **A.** `file -default ug+rw, o+r`
- **B.** `chmod -default 664`
- **C.** `umask 0022`
- **D.** `umask 0002`

7. You've done such a great job of showing how cool Linux is that now there are a bunch of new Linux machines. To make your job easier, you want to consolidate user information in one place, instead of having to make user accounts on each machine. What system, or systems, could you implement to centralize user information?

- **A.** shadow
- **B.** NIS
- **C.** RADIUS
- **D.** LDAP

8. Bob has been assigned to help out the testing_development project and needs to be added to the appropriate group so he can get access to the groups shared files. What command will add Bob to the correct group?

- **A.** `usermod -g testing_dev`
- **B.** `groupmod -u bob`
- **C.** `usermod -G testing_dev`
- **D.** `usermod -Ga testing_dev`

9. You need to take your Linux system offline for maintenance and want to check to see who else may be using it, so you can be courteous and let your users know about it. What command, or commands, will show who else is logged into your system?

A. `w`

B. `finger`

C. `who`

D. `lsof`

10. Your network seems to be running slower than normal, and many users are complaining of odd things happening on their computers. You suspect your company may be the victim of the latest computer virus. What tool could you use to check?

A. `ssh`

B. Nessus

C. Snort

D. Tripwire

11. You need to create a new group to support a new product roll-out. What command, or commands, will let you make a new account?

A. `addgroup project_x`

B. `groupadd -g project_x`

C. `newgroup project_x`

D. `groupadd project_x`

12. You want to tighten security on a particular Linux computer by limiting which users have access to the `sudo` command. Which file should you edit to lock down this feature?

A. /etc/users

B. /etc/shadow

C. /etc/sudoers

D. /etc/passwd

13. You need to set a shared file for read and write access for the file owner and members of the files group and no access for anyone else. Which command(s) will give the desired result?

A. `chmod 440 shared_file`

B. `chmod 660 shared_file`

C. `chmod ug=rw,o=`

D. `chmod og=r,e=`

14. You are testing out SELinux to enhance security on your Linux computer. What mode would you use to let all programs run, but log anything that would fail if you were to lock it down?
    - **A.** *enabled*
    - **B.** *allowed*
    - **C.** *permissive*
    - **D.** *test*
15. You are running out of room on your backup system and want to flag a large temporary file so the tape backup system skips it. What is a way you could do that?
    - **A.** `chmod -s temp_file`
    - **B.** `setattr -d temp_file`
    - **C.** `chattr -d temp_file`
    - **D.** `attr -d temp_file`

SELF TEST QUICK ANSWER KEY

1. **B**
2. **B**
3. **D**
4. **B**
5. **C**
6. **D**
7. **B** and **D**
8. **D**
9. **A**, **B**, and **C**
10. **C**
11. **A** and **D**
12. **C**
13. **B** and **C**
14. **C**
15. **C**

ENDNOTES

[1] Schneier B. Secrets and lies. New York: Wiley and Sons; 2000. p. 84.

[2] Northcutt S, Zeltser L, Winters S, Kent K, Ritchey RW. Inside network perimeter security. 2nd ed. Indianapolis, IN: Sams; 2005. p. 11.

[3] Nemeth E, Snyder G, Hein T. The Linux administration handbook. 2nd ed. Upper Saddle River, NJ: Pearson; 2007. p. 85.

[4] McCarty B. SELinux NSA's open source security enhanced Linux. Sebastopol, CA: O'Reilly Media; 2005. p. 20.

[5] Rogers R. Nessus network auditing. 2nd ed. Boston: Syngress; 2008.

[6] Sotirov A, Stevens M, Appelbaum J, Lenstra A, Molnar D, et al. MD5 considered harmful today, <http://www.win.tue.nl/hashclash/rogue-ca/>; 2008 [accessed 06.25.09].

[7] Burr W. NIST comments on cryptanalytic attacks on SHA-1, <http://www.csrc.nist.gov/groups/ST/hash/statement.html>; 2009 [accessed 06.25.09].

[8] Liu D. Next generation SSH2 implementation: Securing data in motion. Boston: Syngress; 2009.

[9] Morgan A, Kukuk T. Linux-PAM system administrators' guide, <http://www.kernel.org/pub/linux/libs/pam/Linux-PAM-html/Linux-PAM_SAG.html>; 2009 [accessed 06.27.09].

[10] OpenLDAP foundation. OpenLDAP administrators guide, <http://www. openldap.org/doc/admin24>; 2008 [accessed 06.28.09].

[11] OpenLDAP foundation. OpenLDAP administrators guide, <http://www.openldap.org/doc/admin24>; 2008 [accessed 06.28.09].

[12] Kukuk T. The Linux NIS(YP)/NYS/NIS+ howto, <http://www.linux-nis.org/nis-howto/HOWTO/index.html>; 2003 [accessed 06.24.09].

[13] Rigney C, Willens S, Livingston, Rubens A, Merit, et al. RFC2865, <http://tools.ietf.org/html/rfc2865>; 2000 [accessed 06.28.09].

This page intentionally left blank

CHAPTER 11

Troubleshooting and Maintaining Linux

Exam objectives in this chapter

- Monitoring Tools
- Analyzing Logs
- Backing Up and Restoring

UNIQUE TERMS AND DEFINITIONS

- **Compact Disc (CD)** – A 4.72-in. disc developed by Sony and Philips that can store, on the same disc, still and/or moving images in monochrome and/or color; stereo, or two separate sound tracks integrated with and/or separate from the images; and digital program and information files.[1]
- **Digital Versatile Disc**, formerly Digital Video Disc (DVD) – An optical storage medium with improved capacity and bandwidth compared with a CD. DVD, like CD, was initially marketed for entertainment and later for computer users. A DVD can store 4.7 GB (the equivalent of a full-length film with up to 133 min of high quality video) in MPEG-2 format and audio. Additionally, the DVD-ROM drive can read DVD movies, and modern computers with the appropriate hardware or software can decode them in real-time.[2]

- **File Transfer Protocol (FTP)** – A client-server protocol, which allows a user on one computer to transfer files to and from another computer over a Transmission Control Protocol/Internet Protocol (TCP/IP) network. Also, it is the client program used by the user to execute the transfer of files. It is defined in STD 9, RFC 959.[3] There are two modes for FTP: *active* and *passive*.

INTRODUCTION

Ongoing system maintenance is an important part of a system administrator's job. Unfortunately, compared with the fire drills of critical outages or the interesting new challenges of configuring the latest tools, regular maintenance is pretty dull, and is often the first thing that gets skipped when things start to get busy. Stephen Covey would categorize it as "important, but not urgent." It is the sort of thing that, if done well, makes the fire drills much less dramatic, and frees up time that can be spent on other fun projects.[4]

In this chapter, we'll cover common Linux tools for managing a running system, including monitoring performance and logs, and backing up data.

Keep in mind that a little preparation goes a long way. If your system seems to be running slowly or having some other issues, it's very helpful to have some historical data to compare with – so it is advisable to run a few of them now and then under normal load, and maybe save some screenshots so you have a good idea what "normal" looks like. Then you can look for something different if a problem comes up.

MONITORING TOOLS

Linux provides a number of handy tools for reviewing system status and other statistics. Not all of them may be included in a default installation, or be configured to collect data, but they are all supported and can be made to work with a little effort.

Commands

It does not take a fancy, expensive system-monitoring package to pull and compile statistics on how your system is performing. Just about everything you need to measure CPU, memory and disk utilization, I/O performance, and network and disk throughput, among others, are at your fingertips in

utilities that, for the most part, are installed in almost every base Linux system. The usage of the most popular utilities is described in the paragraphs that follow.

sar

The `sar` command (think "system activity report") is a part of the `sysstat` package, if it isn't already on your system. It can be used to gather both current and historic system statistics for the system processor. By default, with no options `sar` displays information collected by a `cron` job that is periodically gathered and stored in files at /var/log/sysstat/sa*, with the last two numbers of the sa* files corresponding to the calendar day the information was collected. Some other useful options include `sar -A` to show *all* statistics, and `sar -n DEV` to show network statistics. You can control the collection of current information by appending two numbers at the end of the command; the first will set the collection interval in seconds, the second tells how many times it should gather the information. For example, `sar -n DEV 5 4` will show network statistics for 5-s time periods a total of four times, then provide a nice average for each interface, as shown in Figure 11.1.

The `sar` command has a very thorough `man` page for further information.

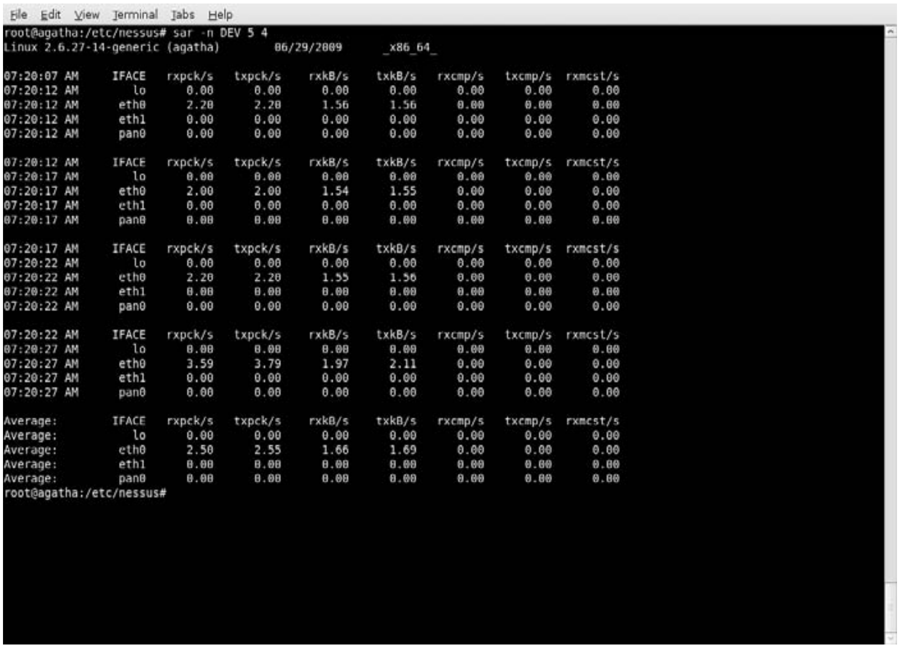


FIGURE 11.1 *An example of using* `sar` *to view network statistics.*

iostat

The `iostat` command shows both CPU and disk utilization statistics, and can also show remote Network File System (NFS) drive systems. It was covered in Chapter 6, "Using BASH," in the "Managing Processes" section.

vmstat

The `vmstat` command uses information in /proc/meminfo, /proc/stat and /proc/*/stat to show virtual memory and other system usage statistics, including disk and processor usage. It is helpful in tracking down potential system bottlenecks. Like the `sar` command, `vmstat` can be followed with two numbers indicating how long to wait and how many times to run. Memory statistics using units of blocks equate to 1024 bytes of memory per block in current Linux kernels. Using `vmstat` without options gives a brief overview of memory and CPU statistics. The `-d` option shows more detailed statistics related to the disk drives in the system. Unlike `top`, `vmstat` doesn't include itself in the statistics that it shows. The man page for `vmstat` is a good resource for further information.

uptime

The `uptime` command gives a quick one-line display showing the current system time, how long the system has been up, the current number of users, and the load average over the last 1, 5, and 15 min. The load average is a count of processes either currently being handled by the processor or waiting to be run.

top

The `top` command shows a current running tally of system usage and an ongoing list of processes. By default, the processes are sorted by the percentage of CPU usage. It was also covered in detail in the "Managing Processes" section of Chapter 6, "Using BASH."

Load Average

You may have noticed that many of the commands that show system utilization display three numbers that represent the system *load average*. You can see it in the upper section of the `top` command, on the end of the `uptime` command, and it also shows up in the `w` command. Because load average is displayed in so many places, you can probably deduce that it's some pretty useful information and should command your attention.

Exam Warning

You may have noticed in the exam objectives that the focus is on commands; however, there are several key statistics that are objectives as well. Load average is one of those statistics. You should be familiar with the commands to know which ones will display load average and other utilization statistics.

As mentioned in the `uptime` section, the load average is a three-part statistic that indicates, on average, how many processes are either currently running or actively waiting to be run by the processor. In his excellent discussion of UNIX load averages, Dr. Neil Gunther summarizes them as "the sum of the number of processes waiting in the run-queue plus the number currently executing."[5] There are slight variations in what gets included in the statistic, but it is essentially a count of how many programs are waiting to be run at a given moment, with a damping factor to average out brief swings to give a more accurate time-weighted average. The 1-min average has a smaller damping factor and is allowed to swing more quickly, whereas the 15-min average has a higher damping factor and gives a better long-term overview of the system load.

Many modern systems have multiple cores and even multiple processors with multiple cores each, which are not factored into calculating the load average. This means that, on a single quad-core processor computer able to handle four simultaneous processes, a load average of two would indicate that (on average, during the time period it question) two of the cores were idle; that is, the system could be considered 50% utilized.

In the real world, some processes will affect a system more than others, and a computer running one set of applications may run fine with a load average up to three or more, whereas another may seem overloaded barely over one. Again, it is helpful to keep an eye on the load average during normal operations to get a feel for what "normal" really is, so that if there is a performance issue, you will recognize a significant change.

If a system starts bogging down and shows high load average numbers, it may be time to find a way to split some of the machine's responsibilities or upgrade to a faster or multiple processor system, provided the application being run will benefit from it.

EXERCISE 11.1: Tracking Down a Runaway Process

Occasionally a program will monopolize an entire system and have to be killed. In this exercise, we'll see how some of the system monitoring tools

can be used to find out what is going on. Occasionally, an errant program will cause such a problem that the entire window manager may become unusable, and it may be necessary to switch to another terminal session completely by using <**Ctrl**><**Alt**><**F2-6**>. From a command prompt, type the following:

1. `top -d 30` is a good command to show which programs are using the most system resources, but the `top` command itself can use a far amount of resources. The `-d` option changes the refresh rate to once every 30 s.
2. `w` is also a good command to show which users are logged in and how much load each is putting on the system.
3. `iostat` may show if a process is waiting for a disk drive that has gone offline or failed. Used with the `-n` option, it will also show if there is a problem with a remote drive connected through NFS.
4. Once you feel the problem process has been isolated, you can use the PID of the problem process from the `top` command with the `kill` command to terminate it. ■

ANALYZING LOGS

Linux, like all UNIX type systems, keeps a record of events in a variety of log files. These files can be crucial for troubleshooting problems, tracking down failures, and finding security issues. Different distributions may have slightly different log files and directory structures. One significant difference is that SuSE Linux uses `syslog-ng` by default, which is a newer, enhanced version of the standard syslog system.

> **Note**
> Although outside the scope of the Linux+ exam, it is worth noting that many network devices support a syslog function to track events. These can include routers, switches, and even some network printers. Your Linux system can be configured to collect all these logs into a single central location to make it easier to sort through and manage them. It is also advisable to have other Linux systems pass their syslog information to a separate machine. If someone should break into your system, they may be able to cover their tracks by altering the local system logs, but they may not be able to break into a separate machine to change the logs there. This can also be useful (in case of hardware failures) to trace a particular fault.

Table 11.1 syslog Event Levels

Level	Approximate Meaning
Emerg	Panic situations
Alert	Urgent situations
Crit	Critical conditions
Err	Other error conditions
Warning	Warning messages
Notice	Thing that might merit investigation
Info	Informational messages
Debug	For debugging only

The syslog configuration is defined in the /etc/syslog.conf file. This file is used to determine which system processes record their events to what files, and what level of logging to use.

System events are categorized into eight levels and can be produced by one of 21 predefined facilities, or system programs. The eight levels, in order of most to least critical, are listed in Table 11.1.[6]

The logging facilities are the various services that might be running on the system, such as mail or File Transfer Protocol (FTP), as well as system events, such as logins, authentications, and `cron` tasks. The catch-all category for miscellaneous events is 'user'.

Common Log Files

The most common log files, the ones covered on the Linux+ exam, are described below. Note that these are defaults that are normally used, but the /etc/syslog.conf file can be used to arrange logging to your particular taste.

/var/log/messages

The /var/log/messages file is the standard location for most system events. It is the place to look to see if your USB flash drive was really recognized by the system and by which drive ID. All info, notice, and warning messages are sent here by default, along with some authorization and cron events.

/var/log/syslog

The /var/log/syslog file is used to collect information about authorization events, time changes, and system statistics events if you installed the *sysstat* package.

/var/log/maillog

If you run a local mail service, events related to it get sent to /var/log/maillog. If your system is a central e-mail server for a busy site, the mail logging may be further broken down into mail.info, mail.warn, and mail.err to help isolate specific levels of messages.

/var/log/secure

The /var/log/secure file is used by Redhat-based distributions to record authorization messages for `sshd`, `sudo`, and other authorization events.

/var/log/lastlog

The /var/log/lastlog is used to store information about user login times. Unlike the other log files, /var/log/lastlog is a binary file, and can't be viewed (well, not properly, anyway) with `vi` like the other log files. Instead, the `lastlog` command uses this file to show the last time a user has logged in to the system. Note that the `last` command is similar in that it shows all the logins from the last time the /var/log/wtmp file has been reset.

Rotating Logs

As a typical Linux system runs, events occur – users connect, programs start and stop, devices are connected and removed, and mail is sent and received, among many other events. All these events get logged. Over time, the logs files keep growing, and if left unchecked would eventually fill whatever hard drive they are written to, causing system problems – and more events to be logged. To prevent this, it is important to implement some sort of log management scheme. Most current Linux distributions include an automatic default log rotation system, scheduled with `cron`.

The standard program for rotating log files is `logrotate`. By default, it is run daily by the `cron` schedule, and can be set to rotate, compress, remove and/or mail log files based on specified times (that is, daily or monthly), or be set to only kick in if a log reaches a specified size. The /etc/logrotate.conf file contains the options used by `logrotate`. More information about the `logrotate` command can be found at `man logrotate`.

Searching and Interpreting Log Files

Most log files are simple text with valuable data that can be read with your favorite paging application, such as `less`. Their value lies in the fact that they tend to be large with lots of different information. While it is both educational and beneficial to browse through them now and then to see what is happening on your computer, if you are searching for a specific type of event or troubleshooting a particular issue, all the other events will simply clutter and get in the way. There are several useful ways to deal with this, include `grep`, `tail`, `awk`, and `sed`.

grep

The `grep` command is an all-around super handy search utility. You can pipe the output of a command into it, and use it to filter what shows up on the screen; or you can feed files directly to it, and it will spit out only the bits you are interested in. The syntax for `grep` is as follows:

```
grep [options] PATTERN [file]
```

Some helpful options are as follows:

- `-i` or `--ignore-case` to ignore the case of the text you are searching for.
- `-v` or `--invert-match` to only show lines that *don't* contain the search pattern.
- `-c` or `--count` will only display the number of lines that match the PATTERN. Used with the `-v` option, it will show the number of lines that *don't* match the PATTERN.
- `-r` or `--recursive` will search down through subdirectories.
- `-a NUM` or `--after-context = NUM` will include NUM number of lines after the matching line.
- `-b NUM` or `--before-context = NUM` will include NUM number of lines after the matching line.

The `before-context` and `after-context` options are handy to provide additional information before and after the matching line to give more data that may be associated with the specific event you are interested in.

One of the handy features of `grep` is that you don't even have to know what file you want to search in:

```
cd/var/log
grep -i usb*
```

Given the above example, each line in every file in the /var/log subdirectory that contains 'usb' in either upper or lower case will be printed on the screen, with the name of the file the line was found in at the beginning of the line.

tail -f

Another very useful utility is `tail`. By itself, `tail` will show the last 10 lines of text in a file; or with the `-n NUM` option, it will show the last NUM lines. Compare this with the similar, but not quite as useful as, the `head` command, which will show the first 10 lines of a text file. The option that really makes `tail` an asset is `-f`, for follow. This option will refresh the screen with new information as the log file gets updated, so you can monitor in real time what the system is reporting. It may be convenient to open a separate terminal window just for running `tail -f`.

Learn by Example: Monitoring System Events Using `tail -f`

The ability to monitor what your Linux system is doing is very helpful. One example is attaching storage devices. Most current systems are very good at automounting storage devices, but now and then it's necessary to mount something by hand, and you'll need to be able to figure out where the storage device is attached to the system. In Figure 11.2, we see a USB flash drive get disconnected and then reconnected to a Linux system. You can see from the figure that the drive is detected as a SCSI-emulated device, in this case sdb1. In the example, it was automatically mounted as /dev/sdb1; but now you know where to go looking for the information you'll need, in case you ever need to use `mount` to manually mount a drive.

awk

The `awk` utility is actually a full-fledged programming language specialized for text and string functions. It can be used directly from the command line for simple functions, or scripted to build your own more complex utilities.[7] The general syntax is as follows:

```
awk [options] '{script-text}' filename
```

The script-text can be replaced with a script file by using the `-f` option like this:

```
awk [options] -f script-file filename
```

If the filename is left off, `awk` will use standard input (STDIN), so it can be used to pipe information between programs. Although it can be used to

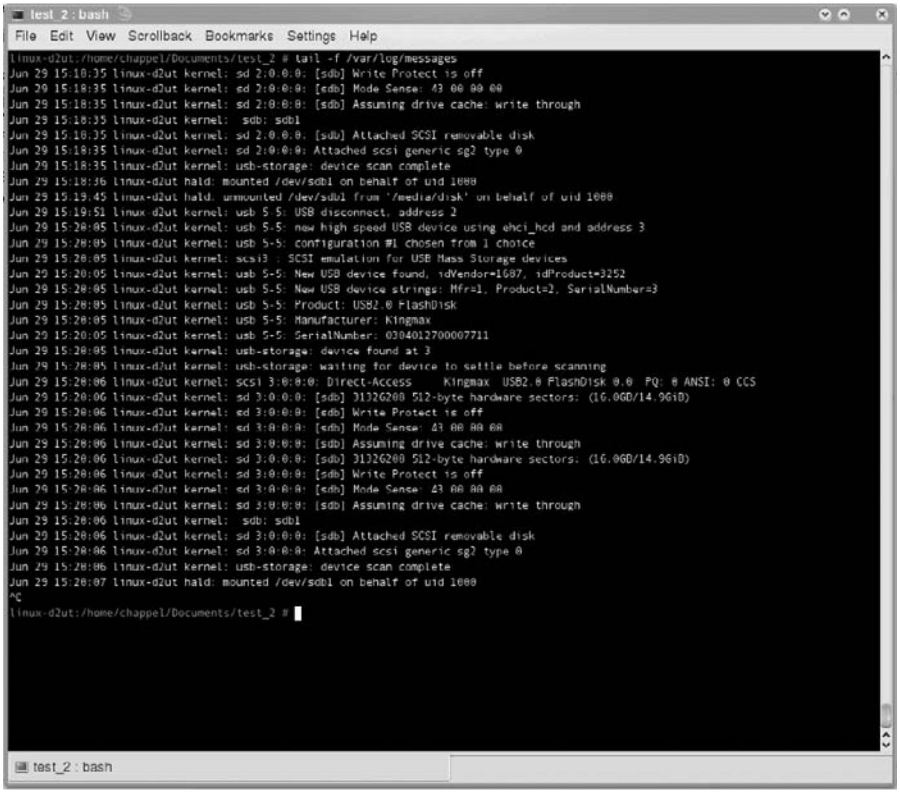


FIGURE 11.2 *An example of* `tail -f` *while a USB flash drive gets attached to a system.*

mimic `grep` to print out lines that contain a search string, `awk` is much more flexible and can be configured to return only parts of lines, and can reorder the output to be used for creating reports based on logs or other files that contain structured text.

sed

The `sed` utility is a *stream editor*, intended to edit files using scripts. It is useful for doing bulk search and replace functions within multiple files, and pretty much anything that you would do with a text editor if you were editing one or two files, but would be tedious for 100.[7]

Like `awk`, it can accept input from a file or STDIN through a pipe from another command, and acts on files on line at a time. The syntax is also similar:

```
sed [options] '{script_text}' filename
```

It can also be used with the meat of the script in a separate file:

```
sed [options] -f script_file filename
```

EXERCISE 11.2: Using `awk`

In this exercise, we'll look at a very basic example of using *awk* to print user names and UIDs from the /etc/passwd file.[8] From a command prompt, type the following:

1. Enter the following command at the command prompt:

   ```
   awk -F:'{print "groupname: "$1",group ID:"$3}'/etc/group
   ```

   Stepping through the example, the `-F`: defines a colon as the field delimiter – it's what separates the different pieces of information in the /etc/group file. The actual script is enclosed in single quotes so the command shell won't try and interpret the squiggly braces. The print command generates the output that shows up on the screen. The *$1* is the first field in the file – in this case the group name – and the *$3* is the third field, the GID. The last part is, of course, the group file that we are searching through.

2. To see the last three lines of /etc/password, enter `tail -n3/etc/password`.

3. To isolate the information on a specific user, you can reenter the command in step (1) (or, if you are using BASH, you can recall the command from your command history by hitting the **up** arrow on the keyboard until you see the command you entered in step (1), and append a pipe character the `grep` command so that it reads like this):

   ```
   awk -F: '{print "groupname: "$1",
     group ID:" $3}' /etc/group | grep [username in list]
   ```

4. We could continue this exercise to use `sed`, but I do not recommend editing /etc/password. You may want to experiment using a copy of the file in another location, or another file that you create. ■

BACKING UP AND RESTORING

Data backups are a conundrum. Few people deny the importance of good backups, but backup systems are frequently neglected. Backups seem easy

enough – just make a copy of stuff you don't want to lose – but a thorough backup system is incredibly complicated.

There are a number of things to keep in mind when working with backups, including the following:

- What to back up? What information would you miss if something happened to your computer? A much tougher question is "what information would your *users* miss if something happened to *their* computer?" Remember that you may need to restore not just data files, but the entire system, potentially including drive types, partitioning and redundant array of inexpensive disks (RAID) information, and database structure.
- Where to back it up? You'll want your information squirreled away someplace safe from fire, theft, and natural disasters, but conveniently at hand for quick restores. Two copies may be necessary – one local and one off-site.
- When to back it up? More frequent backups require more resources, but any data accumulated between backups is at risk.
- What to do with all the copies of your information? It needs to be cataloged in some manner, so you can quickly retrieve and restore exactly the file you want.
- Don't forget that even though it's called a *backup system*, it's the *restore* that is the point of the entire thing. You need to create and review logs to make sure the backups are really happening, and frequently do test restores of not just files but an occasional entire system so that you know everything works the way it should, and so that you'll be familiar with how to do it when operating in full-crisis mode.

Not all backups are the same. There are two general kinds of backups – *complete backups*, that contain everything that you deem worth being able to restore; and *partial backups*, which contain only things that have changed since the most recent complete backup. Partial backups are further divided into *differential* and *incremental*. A *differential backup* will contain everything that has changed since the last complete backup, where an *incremental backup* copies only everything that has changed since the last partial backup. If your data doesn't change much, incremental backups are a lot faster and use much less space than differential; but to do a full restoration, you'll need to get the last complete backup and *all* of the incremental backups. If you do differential backups you'll only need the most recent complete and the most recent differential.

Backups can also be done at different levels. File-level backups run at the operating system level and are convenient for dealing with individual files, but require a working operating system to restore to, and may not work well if the file is in use when you try and copy it. Some software, typically databases and some e-mail systems, offer online and offline backups. An online backup uses the database or e-mail system to read each record or mailbox and make a copy. This works great for being able to restore a single mailbox. An offline backup requires that the software that manages the database or the e-mail system be shut down, and then makes a copy of the entire data structure, more like a file-level backup. This interrupts user access to the system, but results in a backup that is much better for restoring the entire system in one go. It is also possible to do an offline backup of entire drive systems and computers themselves by booting them from an alternate device. This is usually the best method to backup a computer as a whole because everything gets copied without having to skip any files the operating system may need to keep open while it is running, and is frequently the best way to restore an entire system in one go. Depending on the configuration and intelligence of your backup utility, you may end up copying information that really isn't necessary, like swap space, and it is possible to run into problems if you try and restore an entire system onto new hardware if it differs significantly from the old system.

Next, we'll review some of the tools used for making backups in Linux.

Copying Data

A couple tools for making file-level copies of data over a network are `rsync` and `ftp`. By copying over a network, it is possible to easily send your backups to off-site locations, although bandwidth may become an issue.

rsync

The `rsync` utility is a little different than the others we discuss here. It doesn't just copy data, it synchronizes it. If you use `rsync` on a directory, the other end will be made to be an exact duplicate of the original. Depending on the options used, that can include deleting files that aren't in the source directory, so it should be used carefully.

One of its greatest strengths is that it can read data in blocks, and only copy the portion of files that have changed, called the delta-transfer. This feature means it won't copy an entire large file if only a small part of it has been modified, which can really save on bandwidth.

It can be run either using a remote shell such as `ssh` or be configured to run as a service, using its own TCP socket and transport system. Note that using `ssh` takes advantage of its extensive encryption and security features, which other shells such as `rsh` and the native `rsync` daemon don't have. It can also be used locally on a single machine – which, by default, disables the delta-transfer feature.

The `man` page for `rsync` is extensive, and is a good reference when preparing for the exam and for "real life" Linux administration once you have passed the exam. The basic syntax for moving files using `rsync` is displayed in Table 11.2.

There are an astounding number of options available in `rsync`; the `man` page for version 3.0.5 is nearly 2700 lines long. Many of them have very subtle differences and interactions with other features. It may be easiest to start with a demonstration.

If you start with a directory that looks like this:

```
chappel@lavie:~/test$ ls -l

-rw-r--r-- 1 root root 188 2009-07-01 11:13 rights_test
-rw-rw---- 1 chappel chappel 4662 2009-07-01 11:13 rights_test2
-rwxr--r-- 1 chappel chappel 20 2009-06-18 22:54 test.sh
```

If you then perform a basic `rsync` command, the syntax will look like this:

```
rsync * ../rsync_test
```

Table 11.2 `rsync` Command Syntax

File Operation	`rsync` Command Syntax
Copy files locally from one directory to another	`rsync [OPTION...] SRC...[DEST]`
Pull files through remote shell	`rsync [OPTION...] [USER@]HOST:SRC...[DEST]`
Push files through remote shell	`rsync [OPTION...] SRC...[USER@]HOST:DEST`
Pull files through rsync daemon (option #1)	`rsync [OPTION...] [USER@]HOST::SRC...[DEST]`
Pull files through rsync daemon (option #2)	`rsync [OPTION...] rsync://[USER@]HOST[:PORT]/SRC...[DEST]`
Push files through rsync daemon (option #1)	`rsync [OPTION...] SRC...[USER@]HOST::DEST`
Push files through rsync daemon (option #2)	`rsync [OPTION...] SRC...rsync://[USER@]HOST[:PORT]/DEST`

The default `rsync` command will create a new directory for you, called *rsync_test*, the contents of which will be:

```
chappel@lavie:~/test$ ls -l./rsync_test

-rw-r--r-- 1 chappel chappel 188 2009-07-01 13:26 rights_test
-rw-r----- 1 chappel chappel 4662 2009-07-01 13:26 rights_test2
-rwxr--r-- 1 chappel chappel 20 2009-07-01 13:26 test.sh
```

Notice that the files were copied, including the permissions, but the owner and group of the first file was changed to the default owner and group of the user invoking the `rsync` command, and the timestamps were all changed to the time that the time and date that the command was run. This wouldn't matter if you are just backing up your own files; but if you are copying a home directory for a number of users and then restored the files back to all be owned by root, you would have a mess. To avoid that, you can use the `-a` option (for archive), which will preserve additional file information, and run the command with root privileges to preserve the ownership:

```
sudo rsync -a * ../rsync_test
```

Using the `-a` option now gives:

```
-rw-r--r-- 1 root root 188 2009-07-01 11:13 rights_test
-rw-rw---- 1 chappel chappel 4662 2009-07-01 11:13 rights_test2
-rwxr--r-- 1 chappel chappel 20 2009-06-18 22:54 test.sh
```

Additional options that may be helpful are as follows:

- `-r` or `--recursive` to copy the contents of subdirectories
- `-v` or `--verbose` to get extra information during the copy
- `-n` or `--dry-run` to not actually do anything, just show what would be done – very helpful if you are moving large amounts of data, or including options to delete files
- `-z` or `--compress` to use compression while transferring files, which can save bandwidth

File Transfer Protocol

Another common way to transfer files is using FTP. Although not normally used for backups, FTP has been around for a very long time and is well supported. Not that the nearly identical functionality and enhanced security offered by Secure File Transfer Protocol (SFTP) (built on *ssh* – see the "SFTP"

section in Chapter 10: "Securing Linux") will probably mean it will be the more commonly used file transfer method in the future.

Using FTP requires both a client and a server that the client connects to. By default, FTP uses TCP port 21 to establish an initial connection with the server. If the client is using *active mode*, the actual files will be sent from the server to a new separate TCP port on the client. This can cause problems with some firewalls and address translation devices, so it is sometimes necessary to use FTP in *passive mode*, where the client again initiates the secondary data connection.

It is important to know the additional ports that need to be opened to enable active mode FTP. The process for establishing a connection will help to illustrate this. In active mode FTP, the client connects from a random unprivileged port (1023 and above) to TCP port 21. This is called the FTP server's *command port*. The FTP client then starts listening on the port it just used to connect+1. For example, if port 1026 was used to make the initial connection, the FTP client will listen on port 1027. Once the acknowledgement (ACK) has been received by the client, using our example the client sends the FTP command *PORT 1027* to the FTP server. The server will then connect back to the client's port 1027 from port 20, its local data port. To support active mode FTP, the following TCP ports need to be opened:

1. FTP server's port 21 from anywhere (Client initiates connection).
2. FTP client's port 21 and ports 1023 and above (Server responds to client's control port).
3. FTP server's port 20 and ports 1023 and above (Server initiates data connection to client's data port).
4. FTP client's port 20 from ports 1023 and above (Client sends ACKs to server's data port).

Another pair of modes that causes problems are *binary* and *ascii*. When in binary mode, a file is transferred exactly as it is, bit-for-bit. This is necessary if you are transferring a nontext file. In ascii mode, the system assumes the file you are transferring is written using standard ascii-encoded text characters and formatting codes (tabs, returns, line feeds, and so forth). Since these codes vary from system to system, ascii mode tries to convert the codes to what your local system understands; but if the information isn't really ascii, the file may get scrambled. If in doubt, use binary.

When connecting to an FTP server, you can use an account with a username and password; or, if the server is configured to allow it, you can log in as *anonymous*, which usually asks you use an e-mail address as a password.

Use the following syntax to connect to an FTP server:

```
ftp [username[:password]@]ftp_servername
```

If you don't include the username and password, you may be asked for them as a part of the login process.

Once you are connected, you can use the `status` command to show the information about your connection, binary, or ascii modes. Typing `passive` will toggle the active or passive mode, and show the mode you've just switched to; typing `ascii` or `binary` will switch between those modes.

Typing a `?` will show a list of commands. The `ls` command works like you would expect, but it shows the files on the server end. Use `put` to copy a file from your local machine to the server, and `get` to copy a file from the server to your local machine.

> **Note**
> You'll need to be familiar with using FTP from the command line, but for a GUI experience most modern Web browsers support FTP by typing *ftp://ftp_servername* where you would normally type *http://web_servername*.

Archiving and Restoring Commands

The following commands are the standard utilities for backing up data on Linux and other Unix systems. They each have their own strengths and weaknesses.

cpio

The `cpio` program uses binary archive files. It has three basic modes:

- *copy-out* mode creates an archive
- *copy-in* mode reads from an archive
- *copy-pass* mode transfers files from on place to another without creating the actually archive as a middle step.

In *copy-out* mode, `cpio` accepts a list of files that you want to put into your archive from STDIN and sends the output to stdout, so use putting *cpio* to use requires some command line redirection. You can use `ls`, as in `ls | cpio -o > archive_file`, but it is much more common to use `find`, since it performs more flexible searches. The output can be a regular file, device file, or network location.

In *copy-in* mode, `cpio` reads an archive file from STDIN and spits out the files into the current directory. The specific files to be extracted from

the archive can be selected by using a pattern, where `pattern` is a standard filename wildcard. The following commands demonstrate how this is done using `cat` and `cpio`:

```
cat archive_file | cpio -i ''pattern''
```

Using of the `-t` option will generate a table of files contained in the archive without actually doing anything with the files.

For further information consult the man or info pages for `cpio`.

tar

Hearkening back to Chapter 7, "Installing Applications," you may recall that the `tar` was discussed in the context of unarchiving source code to compile an application for deployment. This venerable command has been around for literally decades, and was used on UNIX systems for backing up to and restoring files from magnetic tape. Its name is derived from a shortened version of its function, "Tape Archive." Its purpose is to store files in and extract files an archive file, cleverly called a *tarfile*. This tarfile may be created and stored on any rewriteable medium, such as a tape drive or hard disk.

The following is the syntax, including all available options, for creating a tarfile with `tar`:

```
tar c [bBeEfFhiklnopPqvwX [0-7]] [block] [tarfile] [exclude-file]
   {-I include-file|-C directory|file|file}
```

The more commonly used options for `tar` are listed below in Table 11.3:

Table 11.3 Commonly Used Options for `tar`

Option	Description
`-A`, `--catenate`, `--concatenate`	Append tar files to an archive
`-c`, `--create`	Create a new archive
`-d`, `--diff`, `--compare`	Find differences between archive and file system
`--delete`	Delete from the archive
`-r`, `--append`	Append files to the end of an archive
`-t`, `--list`	List the contents of an archive
`-u`, `--update`	Only append files that are newer than copy in archive
`-x`, `--extract`, `--get`	Extract files from an archive

EXERCISE 11.3: Backing Up Your Home Directory with *tar*

In this exercise, we will use the `tar` command to create an archive of your home directory and extract the files to another location.

1. Enter `cd ~` to navigate to your home directory.
2. Create the tarfile. Enter the following command: `tar -cvf myhome .tar*`
3. Enter `ls -l` and verify that myhome.tar is there. You can verify the contents of the tarfile by entering `tar -t myhome.tar`.
4. Navigate to the /tmp directory by entering `cd /tmp`.
5. It is time to extract the files from your newly created tarfile in /tmp. Enter `tar -xvf myhome.tar`. You do not need to specify a target directory because you are in your target directory.
6. Enter `ls -l` and verify that the contents of *myhome.tar* have been extracted properly.

In this exercise, we used the `v` option (verbose) for all operations. This option is not required; we used it to illustrate what happens behind the scenes as the `tar` command executes. You can increase the level of detail by adding more `v`'s. Try running through the exercise again, using three `v`'s instead of one; for example, `tar -cvvvf myhome.tar*`. ■

The `tar` command is very mature and robust; and as you can see above in the sample syntax for creating a tarfile, there are numerous options for just about every type of file operation. I recommend consulting `tar's man` page and practicing creating, compressing, verifying, and restoring from tarfiles. `tar` is one of those commands that you will use frequently.

dump

Where `tar` and `cpio` are terrific for working with individual files, the `dump` command's sweet spot is archiving entire filesystems. `dump` works by examining files on a target filesystem and determining the files that need to be backed up. These files are then copied to the backup medium of choice, usually hard disk or tape. If the dump is larger than the backup medium, the dump is split and copied to multiple volumes. On most media, the size is determined by writing until an end-of-media indication is returned.

The following is the basic syntax for `dump`:

```
dump -0 -A [archive file] -f [destination file or device]
  [mountpoint of a filesystem ,or a list of files and
  directories to be backed up]
```

In the above-mentioned example, the `-0` option tells dump to perform a full backup. The `-A` option is used to designate the archive file, which is read by the `restore` command (described below) when restoring the backed up files. `-f` is used to identify the target file or device that will host the "dumped" files. The number of options for `dump` is astounding, and I recommend that you consult the `man` page if you need to perform specific types of backups.

restore

The `restore` command performs the inverse function of `dump`. It restores files that have been backed up using `dump`. The `restore` command can be used to restore a full backup of a file system and apply any subsequent incremental backups. `restore` can also be used to restore individual files or directories from a full or partial backup file archive. Like its counterpart, `restore` can operate across a network to restore filesystems or files and directories on remote systems.

The following is the basic syntax for `restore` that will restore all files from the identified archive (mounted at /dev/nst0) in the current directory:

```
restore rf /dev/nst0
```

In this example, the *r* option retrieves all files from the archive and the `f` option is used to designate the archive (/dev/nst0). `restore` also has an interactive mode, accessed using the `-I` option, that lets you navigate inside the archive to select individual files and directories to restore. The syntax for that looks like this:

```
restore if /dev/nst0
```

Like `dump`, the number of options for `restore` is numerous. Its `man` page is worth consulting for performing specific file restore operations.

dd

The `dd` command is a prime example of a big thing that comes in a small package. This seemingly simple tool is incredibly useful for making and copying disk images, backing up and moving disks, and duplicating filesystems. It makes an exact clone of a hard disk, including all blank space. To use it, the source disk must be unmounted and the output destination must be at least as large as the source.

To illustrate, the following syntax will create an image file in my home directory entitled, cdbackup.iso from the CD in my CD-ROM drive:

```
dd if=/dev/cdrom of=/home/brian/cdbackup.iso
```

The option, `if`, designates the input file (the CD that is mounted in my CD-ROM drive) and `of` designates the output file, cdbackup.iso in my home

directory. As you can imagine, the `dd` command provides a very quick way to create an image file that can be used to duplicate a CD or DVD. Creating CDs and DVDs will be described in the next section.

> **Note**
> The `dd` command has earned the notorious nickname, "data destroyer," because system administrators have accidentally destroyed entire filesystems by inverting the parameters of the input file and the output file. If you are using the `dd` command, look over the syntax before you hit the **Enter** key to start the operation. Once started, the operation cannot be reversed.

Writing to Removable Media (CD-RW, DVD-RW)

Creating CDs and DVDs is a relatively commonplace activity that people do for a variety of reasons: backing up files on a computer, creating a photo CD, and transferring music and videos, among others. There are many nifty GUI-based tools to burn CDs and DVDs in Linux. Sadly, the exam does not ask you about any of them. You will be quizzed on how to accomplish this from the command line. At a high level, there are two steps in the process of burning a CD or DVD.

1. Create an image file for the CD or DVD, which involves creating filesystem on the medium and adding data to an image file.
2. Apply the image to the CD or DVD.

Regardless of whether it is a hard disk, USB flash drive, CD or DVD, nothing can be stored on any medium until a filesystem has been created on it. Unlike a hard disk or flash drive, however, the tricky thing about using a CD-R or DVD-R is that it is writeable only once, which means that if you create the empty filesystem as a step by itself, it will remain empty forever and you will have a coaster for your favorite beverage. Burning a CD or DVD involves creating a filesystem while transferring the files to the medium. The command that accomplishes the first part of the process – creating an image file (an .iso file) that includes both the filesystem and the actually files that will be transferred to the medium – is `mkisofs`. You can use the following syntax for creating a typical CD or DVD image:

```
mkisofs -r -o [filename of CD or DVD image.iso]
  [directory where files to be copied are located]
```

The option `-r` sets the permissions of all files on the CD or DVD to be public readable and enables RockRidge-extensions. If you do not use the `-r`

option, the files will be copied with the permissions that were assigned to them in their original locations.

> **Note**
> If you want to create a CD that can be read on Windows, use the `-J` option with `mkisofs` to enable MS Joliet extensions.

The second step is to apply the image to the CD or DVD. This is the stage in the progress where you actually burn the CD or DVD. For this, you need a separate program, `cdrecord`. First, however, you ought to preview the image to make sure that it is everything you expected it to be. Otherwise, you may make a few coasters on the way to achieving what you want. A quick test can be performed by mounting the image file as a local file system and then navigating to the filesystem to have a look around. The following syntax will mount your image file at the mount point, /cdrom:

```
mount -t iso9660 -o ro,loop=/dev/loop0
 [filename of CD or DVD image] /cdrom
```

The `-t` option, iso9660, identifies the filesystem of the image file as that of a CD or DVD. Once the CD or DVD image file is mounted, you can navigate to the /cdrom directory (using `cd /cdrom`) and verify that you have included all of the files you wanted and that the directory structure is sound. If it is not to your liking, you can unmount the filesystem, delete the image file, make the necessary corrections, and start the process again, with no ruined media in the wastebasket or under coffee cups.

With a satisfactory image file, you can now proceed with burning the CD or DVD. First, insert the appropriate blank media in your CD or DVD burner. You will need to figure out the SCSI device address of your CD or DVD burner. As root, issue the command:

```
cdrecord --scanbus
```

The results on your system will resemble the following output:

```
Cdrecord 1.10 (i686-pc-linux-gnu) Copyright (C) 1995-2001
  Jörg Schilling
Linux sg driver version: 3.1.20
Using libscg version 'schily-0.5'
scsibus0:
   0,0,0 0) 'SONY "CD-R CDU928E "1.1n' Removable CD-ROM
   0,1,0 1)*
   0,2,0 2)*
   0,3,0 3)*
```

```
0,4,0 4)*
0,5,0 5)*
0,6,0 6)*
0,7,0 7)*
```

The general syntax for `cdrecord` is:

```
cdrecord [general options] dev=[device address]
   [track options] [filename of CD or DVD image.iso,
   or individual file names]
```

In our example, the device address is 0, 0, 0; therefore, this is the basic syntax for burning the image file to the medium:

```
cdrecord dev=0,0,0 [filename of CD or DVD image.iso]
```

You could include additional options, such as `-eject` to eject the medium after the burn process, or a host of others depending on how much control you want to exert over the process. The `cdrecord` command is part of the `cdtools` suite of tools and has a rather descriptive `man` page that you can consult for further information.

SUMMARY OF EXAM OBJECTIVES

It is fitting that all of the topics covered in this chapter fall at the end of the book. Until this point, you have spent your time preparing your Linux system for deployment in the wild. You have configured your hardware, installed Linux, made it useful with a few applications, and protected your system's users by implementing security measures. This chapter was written to help you with the care and feeding of Linux on your computer. It makes little sense to go through all of the work it took to get your system to this point only to have it become sick and lifeless after a period of use because it was not maintained. Being able to actively monitor the computer, diagnose what is happening (and maybe even being able to prevent a breakdown), and taking care of the system's and your users' valued files are critical activities for maintaining the heath of your system.

There are so many monitoring and measuring utilities available on Linux that there is no excuse not to monitor your system. Although the exam only focuses on the capabilities and the uses of `sar`, `iostat`, `vmstat`, `uptime`, and `top`, in reality, these programs only scratch the surface of what is available. The `sar` and `top` programs will produce detailed reports on how different components of your system are performing at specified intervals and in real time, respectively. You will want to pay special attention to the programs that calculate and display the load average: `top`, `w`, and `uptime`, among others.

Although monitoring will help you check up on how things are going with your system, you will need to know where to go to find information when the inevitable happens and things go badly. Just as there are numerous monitoring programs, there are even more logs. It seems that just about anything that happens in Linux is logged, and these logs, many of which reside under the /var/log directory structure, should be the first things you turn to when starting to troubleshoot. Knowing the various logs under /var/log is important for both the exam and back at your job, especially ./messages, ./syslog, ./maillog, ./secure, and ./lastlog. With the amount of data that is captured in each log, you definitely do not want to have to go line-by-line through a log file to find the information you need. For this, you need good searching and sifting tools; `grep`, `sed`, `awk`, and `tail` are your closest allies.

Mechanical things break and people make mistakes. Restoring data that has been lost due to a hard disk crash or user error is inevitable. In my view, there are two types of network administrator: (1) those who have had to restore lost data from a backup and (2) those who are looking for work because their backups failed and were never tested. The good thing is that Linux offers the network administrator a myriad of methods and tools for backing up and restoring files to a variety of media. This chapter covers the basics for protecting files by ensuring that duplicate copies exist through FTP and `rsync`, and for creating backup sets with `tar` and `dump`. For restoring files, there is `tar` and `restore`. The `dd` command gives you the ability to create disk images and apply them to other media.

SELF TEST

1. Which of the following commands does not display load average?

- **A.** `top`
- **B.** `w`
- **C.** `who`
- **D.** `uptime`

2. What command would you use to generate a static report of CPU utilization?

- **A.** `uptime`
- **B.** `vmstat`
- **C.** `top`
- **D.** `iostat`

3. You want to brag about how long it has been since your server was last rebooted to your colleagues who manage servers that run a different operating system. What is the best command to use to find out how long it has been since your last reboot?

   A. `sar`

   B. `uptime`

   C. `iostat`

   D. `loadav`

4. In the following list, what is not a valid *syslog* event level?

   A. Emerg

   B. Alarm

   C. Err

   D. Notice

5. The performance of your corporate SMTP relay server has been intermittently slow and you suspect a hardware problem. Which of the following log files would you use to look for hardware-related events?

   A. /var/log/messages

   B. /var/log/syslog

   C. /var/log/maillog

   D. /var/log/secure

6. You are trying to get a user's USB flash drive to mount on a Linux workstation and are experiencing trouble. As part of your troubleshooting, you decide that you want to find out if the system is recognizing that the flash drive has been inserted. Using the output of the `dmesg` command as the source, what is the correct syntax for `grep` to find all USB-related events?

   A. `dmesg | grep 'usb'`

   B. `dmesg | grep "USB"`

   C. `dmesg | grep -i 'usb'`

   D. `dmesg | grep -i usb`

7. A user is using FTP to upload a graphics file to a remote server and when the file is loaded in a browser, the browser window is filled with gibberish. What should the user be doing to prevent this from happening?

   A. type `bin` at the FTP prompt to transfer files in binary mode

   B. type `ascii` at the FTP prompt to transfer files in ascii mode

C. type `pasv` to force the FTP connection into passive mode

D. use `mput` instead of `put` to transfer the file

8. Users in your finance department are reporting errors when trying to connect to their server. You decide to monitor activity on this server as the users try to connect. What command would you use with `dmesg` to monitor these system events?

   A. `dmesg | tail -f`

   B. `dmesg | less`

   C. `dmesg | tail -n5`

   D. `dmesg | less -h5`

9. You have been asked to create a CD that contains the personal files of a user who is leaving your company. The files are stored on your Linux-based file server in /home/miranda and the CD needs to be readable on a Windows computer. What is the correct syntax to create the image file for the CD?

   A. `mkisofs -rW -o mirandasfiles.iso /home/miranda`

   B. `mkisofs -rJ -o mirandasfiles.iso /home/miranda`

   C. `mkisofs -r -o mirandasfiles.iso /home/miranda`

   D. `cdrecord -rJ -o mirandasfiles.iso /home/miranda`

10. You are managing Web servers in both a development environment and on the Internet (hostname is www). Once development on a given release is complete and tested, the developers ask you for a solution to keep the content on the staging server in the development environment synchronized with the content on the public Web server. What command would you run on the staging server to ensure that the files and all subdirectories on the staging server are updated on the public Web server as they are updated?

   A. `rsync * www:/home/httpdocs`

   B. `ftp www | put *`

   C. `rsync -r * www:/home/httpdocs`

   D. `ssh www | copy -r * www:/home/httpdocs`

11. You have been asked to back up users' data on a particular server before a core application is upgraded. Because of the amount of data, you need to ensure that these files will fit on a remote hard disk. What command would you use to ensure that the smallest possible size of the backup file?

A. `tar -cvf userdata.tar /home/*`
B. `tar -xjvf userdata.tar /home/*`
C. `tar -cjvf userdata.tar /home/*`
D. `tar -xvf userdata.tar /home/*`

12. You are replacing Michael's computer and have backed up his hard disk to an attached USB external hard disk (/mount/usbhdd) using the following syntax: `dump -0uf -A michaelhdd.archive -f /mount/usbhdd/michaelhdd.backup /`. You want to restore the backup on another hard disk in the new computer. After booting the new computer and mounting the external hard disk, what command do you use?
A. `restore -rf /mount/usbhdd/michaelhdd.backup`
B. `dump -xf /mount/usbhdd/michaelhdd.backup`
C. `tar -xvf /mount/usbhdd/michaelhdd.backup`
D. `restore -rf /mount/usbhdd/michaelhdd.archive`

13. Lately you have been hearing reports that your Linux server is slow to respond, and you have a suspicion that there are applications that are consuming more than their fair share of the server's memory. What key combination would you press while `top` is running so that the running programs are sorted by their respective percentage of memory utilization?
A. **F + M**
B. **F + n**
C. **F + k**
D. **F + l**

14. Users are reporting that a particular corporate server responds slowly for around 30 min between 10:30 A.M. and 11:00 A.M. You decide to run *sar* at regular intervals during this time to capture statistics on the server's network performance. What syntax would you use to capture six sets of these metrics every 10 min?
A. `sar -A DEV 600 6`
B. `sar -n DEV 600 6`
C. `sar -n DEV 6 600`
D. `sar -A DEV 6 600`

15. You are in the process of setting up an active mode FTP server. Whenever you try to connect, you can connect to the server, but you cannot enter a username and password. You made sure that TCP ports 21 and 20 are open on the server. What is the most probable case of the problem?
    - **A.** TCP ports 1022 and below are open on the server
    - **B.** TCP ports 1023 and above are open on the server
    - **C.** TCP ports 1022 and below are closed on the server
    - **D.** TCP ports 1023 and above are closed on the server

SELF TEST QUICK ANSWER KEY

1. C
2. D
3. B
4. B
5. A
6. D
7. A
8. A
9. B
10. C
11. C
12. D
13. B
14. B
15. D

ENDNOTES

[1] Compact Disc, <http://foldoc.org/cd>; 2009 [accessed 07.23.09].

[2] Digital Versatile Disc, <http://foldoc.org/dvd>; 2009 [accessed 07.23.09].

[3] File Transfer Protocol, <http://www.foldoc.org/ftp>; 2009 [accessed 07.23.09].

[4] Covey S. The 7 habits of highly effective people. New York: Fireside Books; 1990 [chapter 5].

[5] Gunther. Dr. N. UNIX Load Average Parts 1 and 2. <http://www.teamquest.com/resources/gunther/display/5/index.htm>; 2003 [accessed 06.29.09].

[6] Nemeth E, Snyder G, Hein T. The Linux administration handbook. 2nd ed. Pearson; 2007. p. 212.

[7] Dougherty D, Robbins A. sed & awk. 2nd ed. O'Reilly Media.

[8] Robbins D. Common threads: Awk by example, Part 1, <http://www.ibm.com/developerworks/library/l-awk1.html>; 2008 [accessed 06.29.09].

Appendix: Self Test

CHAPTER 2: INSTALLING LINUX

1. Your manager has asked you to order the next set of workstations for the department. In addition, the organization has decided to migrate from a Microsoft Windows XP operating system to a Linux operating system environment. As a result, the workstations you order must support a Linux operating system. To verify that the workstation you plan on ordering is supported by the Linux distribution you would like to install, what should you do?

   A. Configure the workstation to dual boot both Windows 98 and Linux operating systems.

   B. Tell your manager that Linux is an operating system for servers only.

   C. Review the HCL for the Linux distribution you would like to install to verify the version of Linux you plan on installing supports the workstations you want to procure.

   D. Check the Microsoft Web site for additional information about installing Windows XP.

   Correct answer and explanation: **C**. Answer **C** is correct because to ensure the version of Linux you are installing functions correctly on your workstation, you need to review the HCL for your system.

   Incorrect answers and explanations: **A**, **B**, and **D**. Answer **A** is incorrect; you are only installing the Linux operating system and you still must verify whether it works on your system via the HCL. Answer **B** is incorrect because Linux operates on workstations and servers and you still must verify whether it works on your system via the HCL. Answer **D** is incorrect; reviewing the Microsoft Web site for installing Windows XP has nothing to do with verifying whether Linux can be installed on your system.

2. Your organization needs a Linux filesystem that supports journaling. Which filesystem supports journaling?

A. ext for VFAT

B. ext2

C. ext3

D. ext5

Correct answer and explanation: **C**. Answer **C** is correct; ext3 does support journaling.

Incorrect answers and explanations: **A**, **B**, and **D**. Answers **A** and **D** are incorrect because ext for VFAT and ext5 are not filesystems (they do not exist.) Answer **B** is incorrect because ext2 does not support journaling.

3. Your organization has decided to implement RAID 5. What is the minimum number of hard disk drives required to support RAID 5?

A. Zero disk drives are required. RAID 5 does not exist.

B. Two disk drives are required.

C. Three disk drives are required.

D. One disk drive and a Tape Backup system are required.

Correct answer and explanation: **C**. Answer **C** is correct because RAID 5 uses three disk drives to perform data striping and distributed parity striping.

Incorrect answers and explanations: **A**, **B**, and **D**. Answer **A** is incorrect; RAID 5 can be implemented within a Linux environment. Answer **B** is incorrect because RAID 5 requires a third disk to implement distributed parity striping. Data Striping (RAID 0) and Mirroring (RAID 1) require only two disks to function. Answer **D** is incorrect; this is not a valid RAID implementation.

4. Which protocol does not support the installation of Linux across a network?

A. HTTP

B. NFS

C. FTP

D. USB

Correct answer and explanation: **D**. Answer **D** is correct; USB is not a protocol. It is an Industry Bus Architecture specification used to establish data transfer between a device attached to your workstation.

Incorrect answers and explanations: **A**, **B**, and **C**. Answers **A**, **B**, and **C** are incorrect; all three are protocols used to install Linux across a network.

5. When installing a Linux distribution source across a network, which network protocol should you use for anonymous login support?

A. SMTP

B. FTP

C. Telnet

D. LDAP

Correct answer and explanation: **B**. Answer **B** is correct; the File Transfer Protocol supports anonymous logins.

Incorrect answers and explanations: **A**, **C**, and **D**. Answers **A**, **C**, and **D** are incorrect; all three are protocols used for different purposes. SMTP is the Simple Mail Transfer Protocol used to exchange e-mail. Telnet is a remote terminal emulation protocol used to access remote terminals on other systems. LDAP is the Lightweight Directory Access Protocol used to authenticate users within a Directory Services environment.

6. Which graphical user interface is supported by the Linux operating system?

A. KDDE

B. GNOOME

C. KDE

D. GMONE

Correct answer and explanation: **C**. Answer **C** is correct because the K Desktop Environment (KDE) is a popular user interface. Another popular interface is GNOME.

Incorrect answers and explanations: **A**, **B**, and **D**. Answers **A**, **B**, and **D** are incorrect; all three are not user interfaces and do not exist.

7. What is the maximum number of primary partitions supported on a hard disk drive for a PC-based system?

A. Five primary partitions are supported.

B. A hard disk drive cannot support primary partitions.

C. Four primary partitions are supported.

D. Only secondary partitions are supported.

Correct answer and explanation: **C**. Answer **C** is correct. Four primary partitions are imposed as a system BIOS limitation for PC-based systems.

Incorrect answers and explanations: **A**, **B**, and **D**. Answers **A**, **B**, and **D** are incorrect; all three are false statements.

8. To perform an HTTP-based network installation, you must enter the following information to establish connectivity with a remote network server.

A. Your workstation IP address and e-mail address.

B. The remote network server's IP address and e-mail address.

C. The remote network server's IP address and remote network server directory containing the Linux distribution source.

D. The remote network server's IP address and your local workstation's directory containing the Linux distribution source.

Correct answer and explanation: **C**. Answer **C** is correct. The Linux distribution source is located on the remote network server. As a result, the target machine must enter the remote network server's IP address to reach the correct server and the directory on the remote network server containing the Linux distribution source.

Incorrect answers and explanations: **A**, **B**, and **D**. Answers **A**, **B**, and **D** are incorrect; all three are false statements. Answer **A** implies that the Linux distribution is located locally on your target machine. If this were the case, you would not need to establish remote network connectivity. Answer **B**, request for an e-mail address, does not indicate where the Linux distribution resides on the remote network server. Answer **D** is a spin-off from the incorrect answer **A**. Again, if the target machine can access the Linux distribution locally, why connect across a network to a remote server.

9. Your organization's management team has decided to implement virtual partition technology. What is the name of the technology within a Linux operating system that supports virtual partitions?

A. Virtual File Transfer (VTP)

B. Logical Virtual Management

C. Disk Mirroring System (DMS)

D. Logical Volume Management (LVM)

Correct answer and explanation: **D**. Answer **D** is correct. Logical Volume Management (LVM) is the correct name of the technology to create logical partitions. This technology is used to remove the

disk space constraints imposed by the standard partitioning disk approach.

Incorrect answers and explanations: **A**, **B**, and **C**. Answers **A**, **B**, and **C** are incorrect; all three are fictitious names for the virtual partitioning technology.

10. What are the extended partitions used for on hard disk drives?

A. To further divide a hard disk drive into smaller partitions.

B. Extended partitions are not supported on hard disk drives.

C. Linux does not support extended hard disk drives.

D. Primary partitions and extended partitions cannot coexist on the same hard disk drive.

Correct answer and explanation: **A**. Answer **A** is correct. Extended partitions are used to get beyond the primary partition limitations imposed by the system BIOS. Extended partitions are used to further divide a hard drive into smaller partitions.

Incorrect answers and explanations: **B**, **C**, and **D**. Answers **B**, **C**, and **D** are incorrect; all three are false statements.

11. When using the `mkfs` command, what is the `-t` option used for when inserted as a parameter?

A. The `-t` option is used to test the network bandwidth.

B. The `-t` option is used to terminate the operating system.

C. The `-t` option is used to assign filesystems to partitions.

D. There is no `-t` parameter associated with the `mkfs` command.

Correct answer and explanation: **C**. Answer **C** is correct. The parameter to assign a file system using the `mkfs` command is `-t`. For example, `mkfs -t ext3 /dev/sda`

Incorrect answers and explanations: **A**, **B**, and **D**. Answers **A**, **B**, and **D** are incorrect; all three are false statements.

12. To see all the current disk drives on your system and the current disk geometry, what command should you enter?

A. `mkfs -t`

B. `flpart -l`

C. `fdisk -l`

D. `diskgeo -t`

Correct answer and explanation: **C**. Answer **C** is correct. The `fdisk -l` command is used to view the current disk drives and the current disk geometry.

Incorrect answers and explanations: **A**, **B**, and **D**. Answer **A** is incorrect; `mkfs -t` is the command used to assign a file system to a disk partition. Answers **B** and **D** are fictitious Linux commands.

13. What is the purpose of the `parted` command?

A. To reclaim unused disk space

B. To establish disk striping

C. To implement RAID 5

D. To test system's on-board memory for defects

Correct answer and explanation: **A**. Answer **A** is correct. The `parted` command is used to reclaim unused disk space on a disk partition.

Incorrect answers and explanations: **B**, **C**, and **D**. Answers **B** and **C** are performed after the space has been reclaimed. **D** is a system procedure that is performed by using the memtest86 application.

14. You are installing Linux on your organization's server. This is a new installation. You must partition the hard disk for the new Linux installation. Which is the best hard disk partition architecture for supporting *root*, *swap*, and *home* partitions?

A. Primary partition architectures should be used for the *root* and *swap* partitions and extended partition architecture should be used for the *home* partition.

B. The *root*, *swap*, and *home* partitions should all be extended partitions.

C. The *root* and *home* partitions should be placed on extended partition architectures and the *swap* partition should be placed on the primary partition.

D. Only *swap* and *home* should be placed on the primary partition and the *root* partition should not be used.

Correct answer and explanation: **A**. Answer **A** is correct. The primary partition should be imposed in the *root* and *swap* partitions to place constraints and boundaries around the partition. This approach would prevent data from expanding one logical partition and contaminating another logical partition. Partition contamination due to data system growth for *root* and *swap* should be restricted to prevent the partitions from flowing over into other partitions. The *home* partition should be extended to allow for growth as a user's home directory grows or more users are added to the system. In addition, for an operating system to boot, one of the partitions must be a

primary partition to support to store the system's boot software and operating system files.

Incorrect answers and explanations: **B**, **C**, and **D**. Answers **B**, **C**, and **D** are incorrect; all three are false statements because the system will never be bootable. In each case, the root partition is placed on an extended partition. This approach would prevent the system from booting. The placing of the *swap* partition on an extended partition is permissible, but it is not a good system design.

15. During the initial Linux installation process, which application is used to test your system's RAM for an x86-based CPU architecture?

A. testmemx86

B. memtest86

C. memtestx86

D. memx86test

Correct answer and explanation: **B**. Answer **B** is correct; memtest86 is a stand-alone memory test application for x86-based systems.

Incorrect answers and explanations: **A**, **C**, and **D**. Answers **A**, **C**, and **D** are incorrect; all three are fictitious application names.

CHAPTER 3: MANAGING FILESYSTEMS

1. Which Linux command is used to assign a filesystem to a partition?

A. `filesys`

B. `mkfs`

C. `fsmake`

D. GRUB

Correct answer and explanation: **B**. Answer **B** is correct; `mkfs` is the Linux command used to assign a filesystem to a partition.

Incorrect answers and explanations: **A**, **C**, and **D**. Answers **A** and **C** are incorrect because those are not valid Linux commands. Answer **D** is incorrect; GRUB is the Linux bootloader program.

2. The Network File System uses which registered port?

A. TCP 2049

B. TCP 80

C. TCP 23

D. TCP 25

Correct answer and explanation: **A**. Answer **A** is correct. TCP 2049 is the registered port for NFS. It also supports UDP 2049.

Incorrect answers and explanations: **B**, **C**, and **D**. Answer **B** is incorrect because TCP 80 is the registered port for http. Answer **C** is incorrect because TCP 23 is the registered port for Telnet. Answer **D** is incorrect because TCP 25 is the registered port for *smtp*.

3. What is the purpose of the /root directory?

A. It is the main directory for all files and system partitions.

B. It provides virtual memory space.

C. It functions as the home directory for the root user.

D. It is a sharable read-only directory for all users to access.

Correct answer and explanation: **C**. Answer **C** is correct; /root functions as the home directory for the root user account. The /home directory functions as the home directory for the subdirectories assigned to typical users.

Incorrect answers and explanations: **A**, **B**, and **D**. Answer **A** is incorrect because this location is the root directory. Answer **B** is incorrect; /swap is the partition allocated to virtual memory space. Answer **D** is incorrect; /usr is the shareable read-only directory in accordance with the FHS.

4. What is the role of the /home directory?

A. It is the location for temporary file space.

B. It provides virtual memory space.

C. It functions as the home directory for the typical user.

D. It is a sharable read-only directory to all users to access.

Correct answer and explanation: **C**. Answer **C** is correct; /home directory functions as the location for the placement of user home directories.

Incorrect answers and explanations: **A**, **B**, and **D**. Answer **A** is incorrect because this location is the /tmp directory. Answer **B** is incorrect; /swap is the partition allocated to virtual memory space. Answer **D** is incorrect; /usr is the shareable read-only directory in accordance with the FHS.

5. What does FHS stand for?

A. Free home space

B. Similar to NFS, but works on an Apple MAC

C. File Hierarchy Specification
D. Filesystem Hierarchy Standard

Correct answer and explanation: **D**. Answer **D** is correct; FHS stands for the Filesystem Hierarchy Standard.

Incorrect answers and explanations: **A**, **B**, and **C**. Answers **A**, **B**, and **C** are all fictitious names.

6. Which Linux command is used to attach a separate storage device to an existing directory?

A. `mkmount`
B. `mount`
C. `umount`
D. `fdisk`

Correct answer and explanation: **B**. Answer **B** is correct; `mount` is the Linux command used to attach a separate storage device to an existing directory.

Incorrect answers and explanations: **A**, **C**, and **D**. Answer **A** is incorrect because it is not a valid Linux command. Answer **C** is incorrect; `umount` is the Linux command to unmount a storage device. Answer **D** is incorrect; `fdisk` is the Linux command used to create partitions.

7. What is contained in the /var/log directory?

A. A variation in system device drivers
B. Data as the result of spooling, logging, and system temporary files
C. A sharable read-only directory for all users to access
D. System libraries and packages

Correct answer and explanation: **B**. Answer **B** is correct; /var/log is a directory that contains data as the result of spooling, logging, and system temporary files.

Incorrect answers and explanations: **A**, **C**, and **D**. Answer **A** is incorrect because this location is the slash (/dev) directory. Answer **C** is incorrect; /usr is the sharable read-only directory in accordance with the FHS. Answer **D** is incorrect; /usr/lib is the directory containing system libraries and packages

8. What argument do you use to obtain an easy readable output for the Linux `du` command?

A. `-h`
B. `-i`

C. `-v`

D. (no options)

Correct answer and explanation: **A**. Answer **A** is correct; the `-h` presents the display summary information in an easy-to-understand output format using kilobytes, megabytes, and gigabytes.

Incorrect answers and explanations: **B**, **C**, and **D**. Answer **B** is incorrect because the `-i` option presents display information about inodes. Answer **C** is incorrect because the `-v` option presents verbose information. Answer **D** is incorrect; no options just provide a disk summary.

9. Your manager has asked you to mount a CD disc on the community workstation in the lobby, so that everyone can access it. The CD disc needs to be mounted on the /media/cdplayer directory. Which `-t` filesystem option must you include?

A. `-t` iso9660

B. `-t` iso

C. `-t` iso9000

D. `-t` ext3

Correct answer and explanation: **A**. Answer **A** is correct; the iso9660 option is the correct filesystem format for CDs.

Incorrect answers and explanations: **B**, **C**, and **D**. Answers **B** and **C** are incorrect because both do not exist. Answer **D** is incorrect because ext3 is the filesystem format for local hard disk drives with journaling support.

10. What is another format for DVDs besides the ISO9660 format?

A. */swap*

B. SCSI

C. Universal Disk Format (UDF)

D. SMBFS

Correct answer and explanation: **C**. Answer **C** is correct; DVDs can also support the Universal Disk Format (UDF).

Incorrect answers and explanations: **A**, **B**, and **D**. Answers **A**, **B**, and **D** are all incorrect; they are not optical disc filesystem formats. */swap* is the partition allocated to virtual memory space. SCSI, the Small Computer System Interface, is a set of standards for physically connecting and transferring data between computers and peripheral devices. The Server Message Block filesystem (SMBFS) is

a framework designed to allow workstations access to directory/file shares on a network-based server.

11. You need to use `fdisk` to establish a partition for a new SCSI disk drive you want to add for extra storage space. The original drives are all IDE drives. Which is the correct syntax?

A. `fdisk /dev/SCSI1`

B. `fdisk /dev/IDE`

C. `fdisk /dev/sda`

D. `fdisk /dev/sdb`

Correct answer and explanation: **C**. Answer **C** is correct; the SCSI device notation for the first disk is /dev/sda.

Incorrect answers and explanations: **A**, **B**, and **D**. Answers **A** and **B** are incorrect because both do not exist as valid default Linux device names. Answer **D** is incorrect; the SCSI device notation for the second SCSI disk is /dev/sdb. The system indicated that this is the first SCSI drive added.

12. Which file, when the system initially starts up, will automatically mount filesystems?

A. /etc/fstab

B. /boot/fstab

C. /dev/devices.map

D. /etc/grub.conf

Correct answer and explanation: **A**. Answer **A** is correct; the /etc/fstab file is used to define and automatically mount filesystems.

Incorrect answers and explanations: **B**, **C**, and **D**. Answer **B** is incorrect because the file does not exist. Answers **C** and **D** are files used during the loading of the Linux kernel by GRUB.

13. What is an ISO loopback device?

A. The transformation of special file into a virtual Linux filesystem

B. A device that returns feedback tests to the monitor

C. The /null driver device

D. The IP address 127.0.0.1

Correct answer and explanation: **A**. Answer **A** is correct; the Linux operating system offers support for an additional unique type of filesystem. This type of filesystem is known as the loopback filesystem. Most Linux distributions have the loopback device compiled

into the kernel. The kernel supports the transformation of a special file containing an image of another filesystem into a device that can be used like any other Linux partition or device. Linux loopback devices are commonly used for CD/DVD ISO images. The disk image created of the CD/DVD disc contains the UDF or ISO 9660 filesystem format. Before accessing the loopback device, the ISO image must be downloaded and mounted. The Linux mount command is used to attach the virtual filesystem image.

Incorrect answers and explanations: **B**, **C**, and **D**. Answer **B** is incorrect because the Linux echo command will return feedback to the monitor. Answer **C** does not exist. Answer **D** is the network adaptor loopback address for testing network connectivity.

14. Which Linux command is used to designate a specific file or partition for swapping?

A. `/swap`

B. `fileswap`

C. `swapon`

D. GRUB

Correct answer and explanation: **C**. Answer **C** is correct; `swapon` is the Linux command used to designate a specific file or partition for swapping.

Incorrect answers and explanations: **A**, **B**, and **D**. Answer **A** is incorrect; */swap* is a precreated `swap` partition created during the initial installation. Answer **B** is incorrect because it is an invalid Linux command. Answer **D** is incorrect; GRUB is the Linux bootloader program.

15. What is the purpose of the Linux `exportfs` command?

A. It functions as the Linux bootloader

B. To partition a storage device

C. To designate a specific file or partition for swapping

D. To activate the access of shared NFS directories

Correct answer and explanation: **D**. Answer **D** is correct; `exportfs` is the Linux command used to activate the access of shared NFS directories on the NFS server.

Incorrect answers and explanations: **A**, **B**, and **C**. Answer **A** is incorrect; GRUB is the Linux bootloader program. Answer **B** is incorrect; `fdisk` is the Linux command used to partition a storage

device. Answer **C** is incorrect; `swapon` is the Linux command used to designate a specific file or partition for swapping.

CHAPTER 4: BOOTING LINUX

1. You need to access your department's Linux server to perform system maintenance. To perform the necessary administrative tasks, all users need to be logged out of the system and they are not allowed to log back into the system while the system maintenance activities are underway. Which runlevel only grants root access?

A. 6

B. 0

C. 2

D. 1

Correct answer and explanation: **D**. Answer **D** is correct because runlevel 1 is used for root level access only in single user mode.

Incorrect answers and explanations: **A**, **B**, and **C**. Answer **A** is incorrect; runlevel 6 reboots your system. Answer **B** is incorrect because runlevel 0 shuts down the system. Answer **C** is incorrect; runlevel 2 allows multiple users to log into the system. However, users are not allowed network connectivity. Login locally is the only option available.

2. Your department's manager would like all Linux users to access their workstations by using a graphical user interface and have network connectivity. Which runlevel uses a graphical user interface by default and grants network connectivity?

A. 2

B. 0

C. 5

D. 1

Correct answer and explanation: **C**. Answer **C** is correct because runlevel 5 grants network connectivity and the system starts up in graphical user interface mode.

Incorrect answers and explanations: **A**, **B**, and **D**. Answer **A** is incorrect; runlevel 2 does not provide network connectivity. Answer **B** is incorrect because runlevel 0 shuts down the system. Answer **D** is incorrect because runlevel 1 is used for granting root level access only in single user mode.

3. Your department's manager would like all Linux users to access their workstations by using a command line mode (no graphical user interface) and have network connectivity. Which runlevel uses a command line mode for multiple users and grants network connectivity?

A. 2
B. 0
C. 3
D. 1

Correct answer and explanation: **C**. Answer **C** is correct because runlevel 3 grants network connectivity and with a command line mode (no graphical user interface).

Incorrect answers and explanations: **A**, **B**, and **D**. Answer **A** is incorrect; runlevel 2 does not provide network connectivity. Answer **B** is incorrect because runlevel 0 shuts down the system. Answer **D** is incorrect because runlevel 1 is used for granting root level access only in single user mode.

4. What is the purpose of the computer system BIOS?

A. Loads the Linux kernel before loading GRUB
B. Allows the user to log into the Linux operating system and change the kernel
C. Presents the biography of Linus Torvalds, creator of Linux
D. Commences the Linux boot process

Correct answer and explanation: **D**. Answer **D** is correct; the system BIOS, after powering up your system, commences the Linux boot process. The system BIOS identifies, tests, and initializes critical system components such as the hard and floppy disk drives, RAM, keyboard, video display card, hard disk, and other hardware. It determines what device will be used to boot the operating system. For this subtask, system BIOS is able to select from various devices for booting (for example, floppy disk drive, hard disk drive, CD/DVD drive) the operating system. The system BIOS selects the first drive and loads the disk geometry characteristics (for example, cylinders, heads, sectors). Finally, it reads the first sector of the boot device to load the Linux bootloader. Then, it transfers the control to the Linux bootloader.

Incorrect answers and explanations: **A**, **B**, and **C**. Answer **A** is incorrect; the Linux bootloader is used to retrieve the Linux kernel. The bootloader is retrieved by the system BIOS. Answer **B** is incorrect

because system BIOS is not a program that interfaces with the Linux User community. Answer **C** is incorrect because the system BIOS does not present the biography of the creator of Linux.

5. You need to access your department's Linux server to perform system maintenance. You need to power down the system to install new hardware components. Which runlevel shuts down your system?

A. 6

B. 0

C. 2

D. 1

Correct answer and explanation: **B**. Answer **B** is correct because runlevel 0 is used to shut down a system.

Incorrect answers and explanations: **A**, **C**, and **D**. Answer **A** is incorrect; runlevel 6 reboots your system. It does not power down, only reboots. Answer **C** is incorrect; runlevel 2 allows multiple users to log into the system. However, users are not allowed network connectivity. Login locally is the only option available. It does not power down your system. Answer **D** is incorrect because runlevel 1 is used for root level access only in single user mode.

6. The Linux servers in your department all use IDE hard disk drives. Your supervisor requested that you reinstall GRUB into the first partition on the IDE first hard disk while the machine is still running. To install GRUB on the IDE hard disk drive's first partition, which shell command should you use?

A. `grub ide`

B. `grub-install /dev/hda1`

C. `grub-install /dev/sda1`

D. `grub-runlevel /dev/hda1`

Correct answer and explanation: **B**. Answer **B** is correct because /hda1 is the correct naming convention for the IDE hard disk drive's first partition. `grub-install` is the command used to install GRUB while the Linux server is stilling running.

Incorrect answers and explanations: **A**, **C**, and **D**. Answer **A** is incorrect because ide is not the proper Linux syntax for the IDE hard disk drive. To install GRUB on a running system, the `grub` shell command is not used. Answer **C** is incorrect because the disk device syntax is for a SCSI Drive. Answer **D** is incorrect; `grub-runlevel` is not a command.

7. You are the Linux system administrator for your IT department. When you normally access your workstation in the morning, you are granted multiuser access with graphical user interface and network connectivity. To perform system maintenance activities, you need to switch runlevels when the system is running. Which command is used to switch runlevels when the system is running?

   A. `runlevels`

   B. `init`

   C. `System Rescue`

   D. `grub`

   Correct answer and explanation: **B**. Answer **B** is correct because `init` is used to change runlevels while a Linux system is running.

   Incorrect answers and explanations: **A**, **C**, and **D**. Answer **A** is incorrect because runlevels is not a command. Answer **C** is incorrect because `System Rescue` is a process that is performed when repairing a corrupted Linux System. This process is accessed when booting up the system from a Linux distribution media. Answer **D** is incorrect because `grub` is used for installing and testing GRUB configuration settings before applying the modifications.

8. How large is the Master Boot Record (MBR) for a hard disk drive with a sector size of 512 bytes?

   A. 1 MB

   B. 512 bytes

   C. 0 bytes

   D. 6 KB

   Correct answer and explanation: **B**. Answer **B** is correct; the MBR is 512 bytes in size. The MBR loads GRUB stage 1. The GRUB stage 1 program uses the first 446 bytes. The remaining 64 bytes are allocated to the partition table for the partitioning of the hard disk drives (for example, Primary partitioning).

   Incorrect answers and explanations: **A**, **C**, and **D**. Answers **A**, **C**, and **D** are incorrect sizes.

9. Which order of events represents the proper Linux boot process?

   A. System BIOS, bootloader, Linux kernel, user logs into the system

   B. Bootloader, system BIOS, Linux kernel, user logs into the system

   C. System BIOS, Linux kernel, bootloader, user logs into the system

   D. User logs into the system, system BIOS, bootloader, Linux kernel

Correct answer and explanation: **A**. Answer **A** is correct; the Linux boot process, more complex than most operating systems, is based on four stages. The four stages are as follows:

- Using system BIOS to run hardware diagnostics and load the bootloader and powering up your system
- Using the bootloader (GRUB) to mount storage devices and load the Linux kernel
- Executing the Linux kernel to mount the root partition and the entire Linux operating system components
- Presenting the user with a Linux Login Screen to all the system

Incorrect answers and explanations: **B**, **C**, and **D**. Answers **B**, **C**, and **D** are incorrect sequence of events for the Linux boot process.

10. Which command is used to send Linux kernel messages to the standard output (for example, computer monitor)?

A. `grub`

B. `dmesg`

C. `init`

D. `kernelprint`

Correct answer and explanation: **B**. Answer **B** is correct; the `dmesg` command is used to send Linux kernel messages to a standard output (for example, computer monitor). `dmesg` accomplishes this by being able to print or control the kernel ring buffer.

Incorrect answers and explanations: **A**, **C**, and **D**. Answer **A** is incorrect because `grub` is used for installing and testing GRUB configuration settings before applying the modifications. Answer **C** is incorrect because `init` is used to execute runlevels. Answer **D** is incorrect because the command does not exist.

11. You are the IT system administrator for the Linux systems in your department. You need to make changes to the default runlevel setting. Which file contains the default runlevel setting?

A. /etc/inittab

B. /etc/grub.boot/inittab

C. /boot/grub/device.map

D. /etc/init.d

Correct answer and explanation: **A**. Answer **A** is correct; the /etc/inittab contains the default runlevel setting. The default setting for runlevel 5 looks like the following.

id:5:initdefault

Incorrect answers and explanations: **B**, **C**, and **D**. Answer **B** is incorrect because the directory path /etc/grub.boot does not exist. Answer **C** is incorrect because /boot/grub/device.map is used to map Linux device names to GRUB device naming conventions. Answer **D** is incorrect because /etc/init.d is the directory containing the runlevels scripts.

12. Your IT department has made several hardware device changes. These changes include modifications to the hard disk drives. You need to make modifications to the GRUB bootloader. Which file should you edit to configure the GRUB stage 2 image?

A. /etc/menu.lst

B. /boot/grub/menu.lst

C. /etc/grub.conf

D. /boot/grub/gurb.conf

Correct answer and explanation: **C**. Answer **C** is correct because /etc/grub.conf contains directory information about the disk partition used to find and load GRUB stage2 image.

Incorrect answers and explanations: **A**, **B**, and **D**. Answer **A** is incorrect because the menu.lst does not reside in the /etc directory and the file is used to determine which operating system is loaded and booted based on **menu** item selected. Answer **B** is incorrect; the file is used to determine which operating system is loaded and booted based on **menu** item selected. Answer **D** is incorrect because the gurb.conf does not reside in the /boot/grub directory.

13. You are the IT system administrator for the Linux systems in your department. You need to make changes to the GRUB device naming conventions. Which file contains the default runlevel setting?

A. /etc/device.map

B. /etc/grub.boot/device.map

C. /boot/grub/device.map

D. /etc/init.d

Correct answer and explanation: **C**. Answer **C** is correct because /boot/grub/device.map is used to map Linux device names to GRUB device naming conventions.

For example:

(hd0) /dev/sda

Incorrect answers and explanations: **A**, **B**, and **D**. Answer **A** is incorrect because the directory path /etc does not contain device.map. Answer **B** is incorrect because the directory /boot/grub.boot does not exist. Answer **D** is incorrect because /etc/init.d is the directory containing the runlevels scripts.

14. The Linux kernel is a critical component in the Linux boot process. Where does it reside on the system?

A. The /kernel directory

B. The /grub/boot/kernel directory

C. The /boot directory

D. The /boot/kernel directory

Correct answer and explanation: **C**. Answers **C** is correct, because the Linux kernel is located in the /boot directory.

Incorrect answers and explanations: **A**, **B**, and **D**. Answers **A**, **B**, and **D** are incorrect; these directories do not exist.

15. The Linux bootloader is a very critical component in the Linux boot process. Where does it reside on the system?

A. It resides in the Master Boot Record.

B. It resides inside the Linux kernel.

C. The /etc directory

D. Inside the system BIOS

Correct answer and explanation: **A**. Answer **A** is correct; the Linux Bootloader resides in the Master Boot Record (MBR). The MBR is the bootloading sector. The system BIOS reads the first sector of the boot device. This sector is 512 bytes in size. For a hard disk drive, this special sector is known as the MBR.

Incorrect answers and explanations: **B**, **C**, and **D**. Answer **B** is incorrect; the Linux bootloader is used to retrieve the Linux kernel. Answer **C** is incorrect because /etc is a folder mounted by the Linux kernel once it obtains control from the Linux bootloader. Answer **D** is incorrect because the system BIOS reads the first sector of the boot device. This sector contains the Linux boot loading program. The system BIOS loads the bootloader into memory and executes the program.

CHAPTER 5: CONFIGURING THE BASE SYSTEM

1. Your manager wants you to change the system prompt for users on their system to reflect that the company has merged with another company and has rebranded itself as Plix. The manger wants users to have the prompt as "plix>"(without the " marks). What is the correct method to undertake this?

   A. Put the following in the /usr/.bashrc file `PS2="plix>"`
   B. Put the following in the /usr/.profile file `PS1="plix>"`
   C. Add the following to the ~/.bashrc file for each user `PS1="plix>"`
   D. Insert the following into the /etc/env file `PS1="plix>"`

   Correct answer and explanation: **C**. Answer **C** is correct as this will display the prompt for the users. Users can also change this prompt to their own setting if they wish to do so later.

   Incorrect answers and explanations: **A**, **B**, and **D**. Answers **A**, **B**, and **D** are incorrect as the configuration file mentioned do not exist in that directory and is not read by the system.

2. You wish to run a program called disp_rights you have developed and placed in your home directory. You have limited its rights so that it can only be executed by yourself. Your username is syngress. Which command will execute the program?

   A. `syngress/disp_rights`
   B. `~/disp_rights`
   C. `~/syngress/disp_rights`
   D. `/usr/home/syngress/disp_rights`

   Correct answer and explanation: **B**. Answer **B** is correct as the ~ symbol specifies your home directory.

   Incorrect answers and explanations: **A**, **C**, and **D**. Answer **A** is incorrect as syngress is the incorrect notation for your home directory. Answer **C** is incorrect as this specifies a subdirectory called syngress off your home directory. Answer **D** is incorrect as this does not specify where your home directory is.

3. You wish to find the default mail server for the domain mycorp.com. Using the `dig` command, you display the information currently held in the DNS server. Which of the resource record types below correctly defines the mail server's IP address?

   A. NS
   B. A

C. MX

D. MS

Correct answer and explanation: **C**. Answer **C** is correct as all mail servers for this zone are specified by the MX records.

Incorrect answers and explanations: **A**, **B**, and **D**. Answer **A** is incorrect as this defines a nameserver in the zone. Answer **B** is incorrect as this defines an IP address stored with a name. Answer **D** is incorrect as this is an invalid record. Answers **A** and **B** may point to the same server as the MX record in Answer **A** but is not the authoritative answer.

4. You are testing the connectivity to a server with an IP address 10.10.10.4. You want to display a continuous output to the screen so that you can see when the remote server is up. The correct command is

A. `ping -c 10.10.10.4`

B. `ping -v 10.10.10.4`

C. `ping -d 10.10.10.4`

D. `ping 10.10.10.4`

Correct answer and explanation: **D**. Answer **D** is correct as this will display a continuous output to the screen.

Incorrect answers and explanations: **A**, **B**, and **C**. Answer **A** is incorrect as the `-c` option requires a count of the number of iterations. Answers **B** and **C** are incorrect as there is no `-v` or `-d` option.

5. There has been a new system installed on your network that you do not know about. You have performed a port scan and can see that ports 80 and 22 are open. From your knowledge of the service ports, this is likely to be which of the following?

A. a Web and FTP server

B. a Web and SSH server

C. a Web and Telnet server

D. a Web and mail server

Correct answer and explanation: **B**. Answer **B** is correct as the normal ports for HTTP and SSH are 80 and 22, respectively.

Incorrect answers and explanations: **A**, **C**, and **D**. Answer **A** is incorrect as FTP is normally found on port 21. Answer **C** is incorrect as Telnet is assigned to port 23. Answer **D** is incorrect as mail is assigned to port 25.

6. You need to change the IP address of an NIC assigned eth0, which has been assigned a static address 10.10.100.45. This is installed in a server in a small company's network. Which of the following will change the IP address most effectively?

   A. Change the /etc/hosts file to reflect the new IP address and reboot the system.

   B. Change the address in the routing table and force a reboot using the command

```
/sbin/route add 10.10.100.45 netmask 255.255.255.0 dev eth0
```

   C. Use the `ifconfig` command to change the address

```
ifconfig eth0 10.10.100.45 netmask 255.255.255.0 up
```

   D. Change the IP address on the module using the following:

```
/sbin/ipchange 10.10.100.45 netmask 255.255.255.0
```

   Correct answer and explanation: **C**. Answer **C** is correct because the `ifconfig` command can be used to change the IP as described.

   Incorrect answers and explanations: **A**, **B**, and **D**. Answer **A** is incorrect as just changing the hosts file will not change the address of the NIC in the system. Answer **B** is incorrect as this adds a route to the routing table and does not change the IP address of the NIC. Answer **D** is incorrect as there is no command called `ipchange`.

7. A user has a computer that does not use DHCP and he/she cannot use hostnames in any command, although IP addresses do work. Where would you look to see how the DN server is defined?

   A. /etc/resolver.conv

   B. /etc/resolv.conf

   C. /etc/hosts

   D. /etc/sysconfig/network

   Correct answer and explanation: **B**. Answer **B** is correct because the DNS server is defined in that file using nameserver.

   Incorrect answers and explanations: **A**, **C**, and **D**. Answer **A** is incorrect as there is no file in that name. Answer **C** is incorrect as the hosts file defines IP address lookups and not the default nameserver. Answer **D** is incorrect as this is the network parameters directory.

8. A user wants to ensure that his wireless card installed correctly in his system only connects to his company's network. This network has an SSID of mycorp and uses WPA2 for added security. How would you ensure this occurs?

A. `iwconfig essid mycorp`

B. `iwconfig default mycorp`

C. `iwconfig default_ssid mycorp`

D. `iwconfig noroam essid mycorp`

Correct answer and explanation: **A**. Answer **A** is correct as the `essid` option specifies the correct SSID to use.

Incorrect answers and explanations: **B**, **C**, and **D**. Answer **B** is incorrect as there is no default parameter. Answer **C** is incorrect as there is no default or `ssid` option. Answer **D** is nearly correct, but there is no noroam parameter to this command.

9. You have just started work as an IT contractor in a company called mycorp. You have been given a laptop with a static address as you manage some of the routers and these have access control lists (ACLs) on them allowing your IP to access them. You want to check that your nameserver, which is currently set on your machine to be 10.10.100.67, is correct. What command would most likely give you the correct result?

A. `traceroute 10.10.100.67`

B. `nslookup 10.10.100.67`

C. `dig dns.mycorp.com`

D. `nslookup dns.mycorp.com`

Correct answer and explanation: **B**. Answer **B** is correct as this will perform a reverse lookup on 10.10.100.67, which you believe is the correct DNS server. This will display the name servers that have been defined – one of which should be this server.

Incorrect answers and explanations: **A**, **C**, and **D**. Answer **A** is incorrect as it will just display a network trace to the server and will not provide any other information. Answers **C** and **D** are incorrect as there is no guarantee that the server dns.mycorp.com exists and therefore these commands may return a "server not found" error.

10. You are working in a medium-sized company and have added a new server to the network. A static IP address of 10.10.100.45 has been assigned to it. The routing table is shown below:

```
Destination  Gateway Genmask       Flags Metric Ref Use iface
127.0.0.0    *       255.0.0.0     U     0      0   0   lo
10.10.100.45 *       255.255.255.0 U     0      0   0   eth0
```

What command will add a default route to 10.10.100.1?

A. `/sbin/route add default 10.10.100.1`

B. `/sbin/defaultroute add 10.10.100.1`

C. `/sbin/route add default gw 10.10.100.1`

D. `/bin/route add default gw 10.10.100.1`

Correct answer and explanation: **C**. Answer **C** is correct as this correctly defines the `route` command to add a default route.

Incorrect answers and explanations: **A**, **B**, and **D**. Answer **A** is incorrect because the parameter `gw` is not included. Answer **B** is incorrect as there is no command `defaultroute`. Answer **D** is incorrect as the normal location of route is /sbin and not /bin

11. You wish to set up the default editor in your environment to be the vim editor instead of the current setting of `vi`. Which would be the best solution to achieve this?

A. Modify the */etc/env.conf* file to set the default editor environment variable `EDIT` to be `vim`

B. Modify the */etc/env.conf* file to set the default editor environment variable `EDITOR` to be `vim`

C. Change the *~/.bashrc* file to set the default editor environment variable `EDIT` to be `vim`

D. Change the */etc/.bashrc* file to set the default editor environment variable `EDIT` to be `vim`

Correct answer and explanation: **C**. Answer **C** is correct as this will change the default editor to be `vim` for yourself.

Incorrect answers and explanations: **A**, **B**, and **D**. Answers **A** and **B** are incorrect as env.conf is not a valid configuration file. Answer **D** is incorrect as this is the wrong location of the .bashrc file.

12. Your employer wants to protect a finance server that has just been installed on their network and has installed ipchains on the system. What is the best description of the ipchains that has been installed?

A. Force external users to log in if they use Telnet to connect to the system.

B. Block all traffic into the system apart from that defined in the */etc/ipchains/allow.conf* file.

C. Accept or deny packets based on the */etc/ipchains.rules* file.

D. Force the system to check all packets entering the system as defined in the */etc/ipchains/ipchains.rules*.

Correct answer and explanation: **C**. Answer **C** is correct because the firewall rules are held in the */etc/ipchains.rules* file.

Incorrect answers and explanations: **A**, **B**, and **D**. Answer **A** is incorrect as ipchains works on the packets and cannot force users to login to a system. Answer **B** is incorrect because there is no allow.conf configuration file. Answer **D** is incorrect as the rules file is in the wrong directory.

13. A user is experiencing connectivity issues with a network port that has been working successfully for a number of months. You have tested the network by connecting another laptop to the same port, which worked. Looking at the hardware, you can see that they have an old NIC and you wish to replace it with a new one. The kernel did not recognize the NIC upon reboot. How would you add the card manually?

A. `modprobe 8139too`

B. `modprobe enable 8139too`

C. `modprobe up 8139too`

D. `add_dep module 8139too`

Correct answer and explanation: **A**. Answer **A** is correct as this is the correct notation for the `modprobe` command.

Incorrect answers and explanations: **B**, **C**, and **D**. Answers **B** and **C** are incorrect as the optional parameters are incorrect. Answer **D** is incorrect as there is no `add_dep`.

14. Which of the following directories is the primary location for the current hardware information of your computer?

A. /sbin

B. /proc

C. /lib/modules

D. /etc

Correct answer and explanation: **B**. Answer **B** is correct because the /proc directory contains the information about the various aspects of a Linux system, including hardware information such as IRQ ports and I/O address for each detected and installed device in the system.

Incorrect answers and explanations: **A**, **C**, and **D**. Answer **A** is incorrect as the /sbin directory contains the administrative binary files and commands such as `ifconfig` reside here. This directory not only includes information about the system but also the commands to manipulate the hardware. Answer **C** is incorrect in that the /lib/modules directory contains most of the hardware driver

module but does not include hardware information such as the IRQ addresses. Answer **D** is also incorrect as the /etc directory contains a number of system configuration files and installation scripts but does not contain all the hardware information.

15. Which of the following configuration files are typically associated to individual user logins with the Bourne shell?

A. ~/.bashrc, /etc/profile

B. ~/.bash_profile, /etc/profile

C. ~/.bashrc, ~/profile

D. /etc/.bashrc, /etc/profile

Correct answer and explanation: **A**. Answer **A** is correct as the /etc/profile configuration file contains the defaults for all users on a global basis. Each user has a .bashrc configuration in his/her home directory.

Incorrect answers and explanations: **B**, **C**, and **D**. Answer **B** is incorrect because not all Linux distributions have the .bash_profile configuration file. Answer **C** is incorrect as the profile configuration file is located in /etc and not the users home directory. Answer **D** is incorrect as .bashrc is associated with global configuration files and is therefore not in the /etc file system.

CHAPTER 6: USING BASH

1. You need to check the configuration file for your Network Time Protocol service. You know that all the configuration files are somewhere in the /etc folder and that the file would be called 'ntp-something' – maybe ntp, ntpd, ntp.config, ntpd.conf…. but you aren't sure. What's the best way to find your configuration file?

A. whereis ntp*

B. ls /etc/ntp*

C. find /etc -file ntp*

D. ls /etc/ntp?

Correct answer and explanation: **B**. **B** is the correct answer because it will show all files in the /etc directory starting with `ntp` followed by anything (or nothing).

Incorrect answers and explanations: **A**, **C**, and **D**. **A** is not correct because `whereis` looks through the current path and is used to locate commands and scripts. **C** is not correct because `-file` is not a valid

parameter; `find /etc -name ntp*` would work, but note that it would have searched through /etc and every subdirectory below it. **D** is incorrect because the ? only substitutes for a single character.

2. You want to show the files in your current directory sorted by date but aren't sure which option for `ls` is correct. How could you find out?

A. `help ls`
B. `info ls`
C. `man ls`
D. `about ls`

Correct answers and explanations: **B** and **C**. **B** and **C** are the correct answers because they will display information about `ls` and its valid options.

Incorrect answers and explanations: **A** and **D**. **A** is incorrect because `help` only works with internal BASH functions; `ls -help` would be correct. **D** is incorrect because `about` is not a valid command.

3. You find a file in your documents directory named *myfile*, but you don't remember what it is. Which is a good way to learn more about it?

A. `test myfile`
B. `check myfile`
C. `file myfile`
D. `info myfile`

Correct answer and explanation: **C**. **C** is the correct answer because the `file` command will show what kind of information is in a file.

Incorrect answers and explanations: **A**, **B**, and **D**. **A** is incorrect because the `test` command is used to evaluate an expression. **B** is incorrect because `check` is not a valid command. **D** is incorrect because the `info` command is used to display usage and option information about commands and programs, not user files.

4. Your Linux machine seems to be running slowly, and you suspect there is a program that is keeping it busy. What is the best way to check for a program that is using a lot of system resources?

A. `top`
B. `iostat`

C. `ps -A`

D. `view`

Correct answer and explanation: **A**. **A** is the correct answer because the `top` program shows current system utilization and statistics for running programs.

Incorrect answers and explanations: **B**, **C**, and **D**. **B** is incorrect because `iostat` shows CPU and disk drive usage but it does not provide any information about specific programs. **C** is incorrect because `ps -A` will show all running processes but does not show how much of the system any processes is using. **D** is incorrect because *view* is used to open a file in *vi* in read-only mode.

5. You have a script called `my_script` you'd like to run every Sunday night at 8:30. You type in `crontab -e` to edit your crontab file. What would the correct entry in your crontab file look like?

A. `8 30 * * Sun my_script`

B. `30 8 * * Sun my_script`

C. `30 20 * * 1 my_script`

D. `20 30 * * Sun my_script`

Correct answer and explanation: **C**. **C** is the correct answer because the crontab format requires the minutes be put in the first column, the hours in 24-h format in the second column, the day of the month (* for any) in the third column, and the month (again a * for any) in the fourth column. The fifth column is the day of the week. Sunday can be represented with either a `1`, `7`, or *Sun*. If the three-character abbreviation is used, case doesn't matter.

Incorrect answers and explanations: **A**, **B**, and **D**. **A** is incorrect because the hours and minutes are reversed, and the hours are not in 24-h format. **B** is incorrect because the hours are not in 24-h format. **D** is incorrect because the hours and minutes are reversed. Don't forget that on a real machine, you should include the path to your script.

6. You just got a new gps-based Network Time Protocol server so you no longer have to mooch off some university over the Internet. You edit the appropriate config file to add the IP address of your server. What do you need to do next?

A. Nothing. The `ntp` service automatically detects the change and will start using your new time source.

B. Use `chkconfig -d ntp` to turn off the service, then `chkconfig -a ntp` to turn it back on with the new settings.

C. Use the init scripts to stop - /etc/init.d/ntp stop – then start - /etc/init.d/ntp start – the service with the new configuration.

D. Use the init scripts to restart the service - /etc/init.d/ntp restart.

Correct answers and explanations: **C** and **D**. Both **C** and **D** are correct because either will stop the service and restart it with the modified configuration, although **D** requires less typing.

Incorrect answers and explanations: **A** and **B**. **A** is incorrect because `ntp`, like other services, doesn't monitor its configuration files. Note that the /etc/init.d/ntp script `addserver` option will automatically start using a new server. **B** is incorrect because `chkconfig` doesn't change the current status of the service; it only tells the system at which runlevels `ntp` should be started or stopped.

7. You are logged into a Linux computer in a windowed (GUI) environment as user jim and open a terminal (console) session. What directory will you start out in?

A. /home/jim

B. /user/jim

C. /jim

D. /pwd/jim

Correct answer and explanation: **A**. **A** is the correct answer because by default all users' home directories are under /home/<username>.

Incorrect answers and explanations: **B**, **C**, and **D**. **B** is incorrect because /user isn't a standard Linux directory. There is a /usr directory, but it is not used for home directories. **C** is incorrect because home directories typically go under the /home folder. **D** is incorrect because *pwd* is a command to show the current active directory, not a folder.

8. You just finished up a project and have all your files in a folder called *project1*. You archive the folder to a CD and now want to delete it from your computer. Which command will get rid of it for you?

A. `rmdir project1`

B. `rm project1/*`

C. `del project1`

D. `rm -r project1`

Correct answer and explanation: **D**. **D** is the correct answer because the `-r` (recursive) flag will remove all the contents of a folder, including subfolders, and then delete the folder itself.

Incorrect answers and explanations: **A**, **B**, and **C**. **A** is incorrect because `rmdir` will only delete empty folders; it will fail if there are any files in the folder. **B** is incorrect because `rm` without the `-r` option will only remove files, not the folder. Note that if there are no subdirectories under *project1*, as in **B**, then **A** will remove both the files and then the folder. **C** is incorrect because `del` is not a valid Linux command.

9. You need to make a quick update to your /etc/hosts file, so you open it in `vi` and make your changes. Now you are ready to save the file and exit. What do you do?

A. type `<Ctrl> c`

B. type `<Ctrl> d`

C. type `<Ctrl> z`

D. type `<Esc> :wq`

Correct answer and explanation: **D**. **D** is the correct answer because an <esc> is needed to switch from Edit to Ex mode, then a: to switch to command mode. The `w` writes the file, and `q` quits the program.

Incorrect answers and explanations: **A**, **B**, and **C**. **A** is incorrect because `<control> c` is used to stop ongoing programs such as `top` or `tail -f`, not `vi`, which will helpfully remind you of the correct syntax instead. **B** is incorrect because `<control> d` means "end of file." It is used to finish the keyboard entry for input from stdin. An example would be the `at` command. In `vi`, it takes you to the end of the file. **C** is incorrect because `<control> z` puts the current program into the background. Although you will get back to a command prompt, `vi` will still be running. Typing `fg` (foreground) will put you back into `vi`, right where you left off.

10. You want to learn more about all the hidden files in your home directory. What command could you use to see them?

A. `ls -A`

B. `ls .*`

C. `ls -a`

D. `ls .`

Correct answers and explanations: **A**, **B**, and **C**. **A**, **B**, and **C** are all correct. **A** is correct because `ls -A` will show all files, including hidden files (files that start with "."). **B** is correct because it will also show all files that start with a ".". **C** is correct because `ls -a` will

show hidden files. Note that `A` and `a` are different options; the `a` option will show the "." and ".." directory shortcuts; *A* will not.

Incorrect answer and explanation: **D**. **D** is incorrect because it will only show the "." (current) directory; it is the same as typing `ls`.

11. You are documenting your system and want a file named *user_dirs* that contains a current list of all the user home directories. What is a quick way of doing this?

A. `echo /home > user_dirs`

B. `ls /home > user_dirs`

C. `ls /home >> user_dirs`

D. `cp /home >> user_dirs`

Correct answers and explanations: **B**. **B** is the correct answer because it will list the contents of the /home directory and that listing will be placed in the *user_dirs* file. Note that **B** will overwrite any existing file. If desired, using *ls -1* will format the output in a single column.

Incorrect answers and explanations: **A**, **C**, and **D**. **A** is incorrect because `echo` will not list the contents of the folder; you will get a file *users_dirs* that contains the text "/home." While the command in **C** will append the directory listing to the end of the user_dirs file, if the information in the file was captured on an earlier date and that information has changed, you will end up with a file that has both current information from the most recent execution of the command and obsolete information from the previous execution. **D** is incorrect because the *cp* command will attempt to copy the /home folder to a new destination. Because there is no destination listed, an error will be generated on the console (because stderr wasn't redirected) and the file *user_dirs* will be created but empty.

12. You are getting hungry and you can't believe it isn't lunch time yet. You want to check that the NTP process is really running and that your computer clock isn't slow. What command will confirm that ntp is running?

A. `ps -A | grep ntp`

B. `ps -C ntp`

C. `ps ntp`

D. `/etc/init.d/ntp status`

Correct answers and explanations: **A** and **D**. **A** will list all running programs, then search for a line that contains "ntp" and show it on

the screen. Note that it will only confirm that the process is running, not if it is synchronized. **D** is correct because the `ntp` init script supports the *status* flag, which will confirm it is running and show the synchronization status. Note that not all init scripts support a `status` option; Answer **A** will work with any service.

Incorrect answers and explanations: **B** and **C**. **B** is incorrect because `ps -C ntp` will look for an exact match for "`ntp.`" The actual service is `ntpd`, so no match will be found. **C** is incorrect because `ps` will interpret the `ntp` as an option, which is invalid.

13. You have a script name some_program that needs to start in 30 min, and you decide to try the `at` command instead of setting the alarm on your watch to remind yourself. What is the correct syntax?

A. `at 30 <return> some_program <return> <ctrl>d`

B. `at now + 30 minutes <return> some_program <return> <ctrl>d`

C. `at 30 minutes some_program <ctrl>d`

D. `at now + .5 hours <return> some_program <return> <ctrl>d`

Correct answer and explanation: **B**. **B** is the correct answer because the keyword *now* understands adding time using the + sign and descriptive time units. The return is necessary to start the input for what command(s) to run, and the <ctrl>d is needed on a separate line to end the input.

Incorrect answers and explanations: **A**, **C**, and **D**. **A** is incorrect because the time parameter is wrong and no units are given. **C** is incorrect because the time format is wrong and the script that is to run is not on a separate line. **D** is incorrect because the time parameter must be given in whole numbers.

14. You are copying a bunch of files from ./temp to ./new_stuff, but accidentally type `cp temp/* new-stuff` instead of `new_stuff`. You've been reading up on the command history function and want to use that to reenter the command correctly. What do you type?

A. `! -change new-stuff new_stuff`

B. `!!s/new-stuff>new_stuff`

C. `  -  _`

D. `history -r new-stuff new_stuff`

Correct answer and explanation: **C**. **C** is correct because the first ^ identifies text to be searched for and the second ^ identifies the text that will be inserted in place of the found text.

Incorrect answers and explanations: **A**, **B**, and **D**. **A** is incorrect because a single ! expects a line number or beginning characters from the history file; it doesn't take options. The syntax in **B** is incorrect; the "greater than" character should be a fore slash, that is, the proper syntax is `!!s/new-stuff/new_stuff`. **D** is incorrect because the history command option `-r` isn't used for replacement.

15. You just made a new script `my_new_script` and you want to run it. You remember to give an explicit path to it when you execute it by typing `./my_new_script`, but you only get an error saying you don't have permission to run the file. You remember that you need to give yourself execution rights to the file. How do you do that?

A. `chmod 744 my_new_script`

B. `chmod 666 my_new_script`

C. `chmod u+w my_new_script`

D. `chmod g+x my_new_script`

Correct answers and explanations: **A**. **A** is correct because it sets the file's permissions to allow read, write, and execute for the owner but only read access for group and everyone else.

Incorrect answers and explanations: **B**, **C**, and **D**. **B** sets user, group, and everyone else's rights to read and write, but not to execute. **C** also adds write permission for the file owner, but not the execute permission, which is the one that is required. **D** adds the execute permission for the files group. Note that even if you are a member of the files group and if you are the owner, you will need user execute rights to execute the file.

CHAPTER 7: INSTALLING APPLICATIONS

1. A user in your finance department has approached you to let you know that an update is available for one of their core applications. It is critical that the installation go as smoothly as possible. You have been asked to perform the upgrade. What is the correct syntax for safely upgrading the existing application?

A. `rpm -uvh`

B. `rpm -ivf`

C. `rpm -Uvh`

D. `rpm -Ivh`

Correct answer and explanation: **C**. Answer **C** is correct because the switch needed to safely upgrade an existing application is -U.

Incorrect answers and explanations: **A**, **B**, and **D**. Answer **A** is incorrect because -u is an invalid switch; the switches are case sensitive and the correct switch is -U. Answer **B** is incorrect because, while the new application package will be installed, it will be forcefully (-f switch) installed over the existing application, which could produce very undesirable results. Answer **D** is incorrect because -I is an invalid switch; the correct switch is -U.

2. You have used the command `tar -czvf work.tar` to compress a tarball. What will the result be?
   - **A.** An archive file compressed with `gzip` and given the `gz` extension
   - **B.** An archive file compressed with `gzip` and given the `tgz` extension
   - **C.** An archive file compressed with `bzip` and given the `bz2` extension
   - **D.** An archive file compressed with `bzip2` and given the `bz2` extension

   Correct answer and explanation: **B**. Answer **B** is correct as this command will compress the tarball with `gzip` and give it the `tgz` extension.

   Incorrect answers and explanations: **A**, **C**, and **D**. Answer **A** is incorrect as `gz` is given to files compressed directly with `gzip` and not using `tar`. Answer **C** is incorrect as `tar` does not use `bzip`. Answer **D** is incorrect as tar does not use `bzip2`.

3. You have downloaded the source files for a program you want to install. You have unpacked the archive and want to see if there are any instructions on how to compile it. What should you look for first?
   - **A.** Look for a file called `configure` in the directory structure.
   - **B.** Look for a file called INSTALL in the directory structure.
   - **C.** Look for a file called FIRST in the directory structure.
   - **D.** Look for a file called `make` in the directory structure.

   Correct answer and explanation: **B**. Answer **B** is correct as a large number of software packages will have a text document called INSTALL bundled with the package to inform the user on how to compile and install it.

   Incorrect answers and explanations: **A**, **C**, and **D**. Answer **A** is incorrect as `configure` is a script that you use to configure the software for your environment. Answer **C** is incorrect as this is not a recognized file to bundle with the software. Answer **D** is incorrect as `make` is used to compile and install the software.

4. You need to add a new local repository to a system that has .deb software repositories. Which file should you edit to achieve this?

   A. /etc/apt/source.list
   B. /etc/apt.d/sources.list
   C. /etc/apt/apt.d/sources.list
   D. /etc/apt/sources.list

   Correct answer and explanation: **D**. Answer **D** is correct as the sources.list file is usually located in the /etc/apt directory.

   Incorrect answers and explanations: **A**, **B**, and **C**. Answer **A** is incorrect as this is not the correct file; it should be /etc/apt/sources.list. Answers **B** and **C** are incorrect as there is no apt.d directory.

5. You want to install a new ftp program onto your desktop. You are currently running version 5.4 release 8 of the software. Which file should you download to upgrade this software to the latest version available?

   A. ftp-6.0-8.i386.rpm
   B. ftp-9-6.0.i386.rpm
   C. ftp-5.4-8.i386.rpm
   D. ftp-6.0.i386.rpm

   Correct answer and explanation: **A**. Answer **A** is correct as this is the latest release (version 6.0, release 8).

   Incorrect answers and explanations: **B**, **C**, and **D**. Answer **B** is incorrect as the format of the file is incorrect. The file format is name-version-release.architecture.rpm. Answer **C** is incorrect as this is version 5.4, release 8, which is the same version as you have installed. Answer **D** is incorrect as the format of the file is incorrect (see answer B).

6. You have a version of Linux that is managing the software packages using *yum*. You are going to remove the MySQL database group that is currently loaded on it so you can install postgreSQL. What is the correct command to use to remove the MySQL database group?

   A. `yum removegroup MySQL`
   B. `yum groupremove 'MySQL database'`
   C. `yum remove 'MySQL database'`
   D. `yum remove --force 'MySQL database'`

   Correct answer and explanation: **B**. Answer **B** is correct because this will remove the entire package group.

Incorrect answers and explanations: **A**, **C**, and **D**. Answer **A** is incorrect as this is an incorrect syntax. Answer **C** is incorrect as this command is trying to remove the whole group as the syntax is wrong. Answer **D** is wrong as the `--force` option is incorrect.

7. You need to compress a series of files as much as possible as you want to put them onto a CD-ROM to send to someone and they are currently much bigger. What option would you use with gzip to achieve this?

A. `-1`

B. `-best`

C. `--9`

D. `--best`

Correct answer and explanation: **D**. Answer **D** is correct as the option `--best` will take the longest time but achieve the highest level of compression.

Incorrect answers and explanations: **A**, **B**, and **C**. Answer **A** is incorrect as this will give the least amount of compression. The compression ratios go from least compression (1) to most compression (9). Answer **B** is incorrect as the option is `--best` and not `-best`. Answer **C** is incorrect as the option is -9 and not --9.

8. One of your servers has had a drive failure and you need to restore the data from last night's backup, which is a compressed tar archive. You generated the archive of all home directories with the commands:

```
cd /home
tar czvf work.tgz home
```

You have copied the file to /tmp on the new drive and execute the command

```
tar xzvf work.tgz
```

from that directory. To what location would the home directories be restored?

A. They would be restored at the original location (/home) on the new drive.

B. They would be restored to the root directory (/).

C. They would be restored to /tmp/home.

D. The *o* option needs to be specified to overwrite the default home directories setup by Linux or the tar command will return an error.

Correct answer and explanation: **C**. Answer **C** is correct as the tar file was produced using relative and not absolute files. If the `tar`

command had been `tar czvf work.tgz /home`, then this would have restored them to the original location.

Incorrect answers and explanations: **A**, **B**, and **D**. Answer **A** is incorrect as the files were archived with a relative and not absolute location. Answer **B** is wrong as the files are restored relative to where you are (/tmp). Answer **D** is wrong as the option *o* is invalid.

9. You are installing a number of new packages to an older machine that does not have a large amount of disk space and you do not want to install any documentation on the packages as you can look at this on another machine. What option should you add to rpm to ensure this happens?

A. Include the option `--minimumsize`

B. Include the option `--excludedocs`

C. Include the option `--excludedocuments`

D. Include the option `--nodocs`

Correct answer and explanation: **B**. Answer **B** is correct as this is the right option to install the software without any supporting documentation.

Incorrect answers and explanations: **A**, **C**, and **D**. Answer **A** is incorrect as there is no `minimumsize` option. Answer **C** is incorrect as the option is `excludedocs` and not `excludedocuments`. Answer **D** is incorrect as there is no `nodocs` option in rpm.

10. You have downloaded the source code for a new Web program into /tmp and have read the INSTALL file that came with it. The INSTALL file says you have to run the command:

```
./configure --prefix=targetdirectory
```

Where do you think the *makefile* will be created?

A. In the current directory

B. In a subdirectory called targetdirectory from the current directory

C. In a directory called targetdirectory in your home directory

D. Nowhere as the *makefile* is created by `make`

Correct answer and explanation: **B**. Answer **B** is correct as the option `--prefix` will change the path where the *makefile* is created.

Incorrect answers and explanations: **A**, **C**, and **D**. Answer **A** is incorrect as this is the normal location of the *makefile*, but it is overridden by the prefix option. Answer **C** is incorrect as the *makefile* would be created here if you located the source files here or added the path into the prefix option. Answer **D** is incorrect as `make` uses the *makefile* produced by configuration.

11. You have downloaded the rpm files for a new program. Assuming you are a normal user, what else must you do to ensure that you can install the programs?

A. Change the owner of the files to everyone and run `rpm`.

B. Run `chmod 777` on the files before executing the `rpm` command.

C. Use the command `rpm -c rpmfile` to ensure that the system prompts you for the superuser password.

D. Run `rpm` as superuser.

Correct answer and explanation: **D**. Answer **D** is correct as *rpm* must be run with superuser privileges.

Incorrect answers and explanations: **A**, **B**, and **C**. Answer **A** is incorrect as you do not need to change the ownership of the files and you need to have superuser privileges. Answer **B** is incorrect as you do not need to change the permissions of the files and you need to have superuser privileges. Answer **C** is incorrect as the `-c` option will not make the system prompt you for a superuser password.

12. You want to set up a local repository on a server in your network. You are using yum and want to ensure that the repositories will work with this. What tool should you use and where should the metadata files be stored?

A. Use `createrepo` and store the metadata in /etc/yum.repo.d

B. Use `create-repo` and store the metadata in /etc/yum.repo.d

C. Use `createrepo` and store the metadata in /etc/yum/yum.repo.d

D. Use `createrepo` and the metadata will be stored automatically in the correct location.

Correct answer and explanation: **A**. Answer **A** is correct as you need to use `createrepo` and then store this data in its own file in /etc/yum.repo.d

Incorrect answers and explanations: **B**, **C**, and **D**. Answer **B** is incorrect as there is not a `create-repo` command. Answer **C** is incorrect as the correct file location is normally /etc/yum.repo.d. Answer **D** is incorrect as `createrepo` will not store the data automatically.

13. Which of the following commands will not upgrade an installed .deb package?

A. `apt-get install package_name`

B. `dpkg -i package_name`

C. `apt-get --reinstall install package_name`

D. `apt-get update package_name`

Correct answer and explanation: **D**. Answer **D** is correct because "`apt-get update`" performs an update of the APT database but does not upgrade any of the installed applications themselves. Furthermore, the syntax is incorrect because a package name is not used with `apt-get update`.

Incorrect answers and explanations: **A**, **B**, and **C**. Answers **A**, **B**, and **C** are incorrect because all these three commands can be used to install or upgrade an installed .deb package.

14. You are compiling source code for installation and you want to string all of the required commands together to run while you are going downstairs to grab a coffee so that the binary file is ready when you return. What answer below has syntax that will not work?

A. `./configure; make; make install`

B. `./configure / make / make install`

C. `./configure && make && make install`

D. `./configure | make | make install`

Correct answer and explanation: **B**. Answer **B** is correct. The fore slash is not a valid character for use in concatenating commands. It is used in denoting filesystem locations, such as the root directory, "/".

Incorrect answers and explanations: **A**, **C**, and **D**. The semicolon and the double ampersands have a similar function; they simply concatenate the commands. The first command executes and once it terminates, the second command executes, and so on. The `pipe` command, "`|`", sends the output of the first command as input into the second command. Please refer to Chapter 6 for information on scripting and output redirection.

15. You are powering up a laptop that has Linux installed that has not been used in a couple of months. Since you will be handing it over to a user who needs to use it on a long trip, you want to ensure that its applications are current. What command do you run once the APT database is up-to-date?

A. `apt-get update`

B. `apt-get upgrade`

C. `apt-get dist-upgrade`

D. `apt-get install`

Correct answer and explanation: **B**. Answer **B** is correct because `apt-get upgrade` uses the information in the APT database,

which is updated using `apt-get update`, to update all installed applications to their most current versions.

Incorrect answers and explanations: **A**, **C**, and **D**. Answer **A** is incorrect because it is the command to update the APT database. Answer **C** is incorrect because the `dist-upgrade` option upgrades the entire distribution to the latest version and the requirements in this question ask for only the installed applications to be updated. Answer **D** is incorrect because, when a package name is specified, it is the command to install an individual application.

CHAPTER 8: INSTALLING, CONFIGURING AS A WORKSTATION

1. A user has sent three jobs to a printer, with job numbers 372, 373, and 374. They now want to remove printer job number 373, which has not been printed yet. Which command will achieve this?

   A. `lpr --cancel 373`

   B. `lpstat -c 373`

   C. `cancel 373`

   D. `lpr -c 373`

   Correct answer and explanation: **C**. The correct answer is **C** as the `cancel` command will cancel the given job provided it is still in the queue and the user is the owner or has the correct permissions.

   Incorrect answers and explanations: **A**, **B**, and **D**. Answers **A** and **D** are incorrect as the `lpr` command submits jobs to a printer. Answer **B** is incorrect as the `lpstat` command will display the status of a printer.

2. A user is running KDE as his display environment. What will be the most likely environment variable for the display manager?

   A. DISPLAYMANAGER="KDE"

   B. DISPLAYMANAGER="kdm4"

   C. DISPLAYMANAGER="kde_display"

   D. DISPLAYMANAGER="gdm"

   Correct answer and explanation: **B**. The correct answer is **B** as this will specify version 4 of the K Display Manager.

   Incorrect answers and explanations: **A**, **C**, and **D**. Answer **A** is incorrect as KDE is the overall K Desktop Environment and not the display manger. Answer **C** is wrong as this does not define K Display

Manager. Answer **D** is incorrect as this refers to the GNOME display manager.

3. A user is sending a job to printer EPSON_COLOR, which is not their default printer. They want it printed with a banner title of *myjob* and then a mail is sent to let them know when it has been printed. The user will need to use the following command:

   A. `lpr -m -T myjob -P EPSON_COLOR`

   B. `lpr -sendmail -C myjob -P EPSON_COLOR`

   C. `lpr -m -T myjob`

   D. `lpr --sendmail -T myjob -P EPSON_COLOR`

   Correct answer and explanation: **A**. Answer **A** is correct as this sends a mail to the user with the `-m` option, prints the title *myjob* on the banner page, and sends it to printer *EPSON_COLOR*.

   Incorrect answers and explanations: **B**, **C**, and **D**. Answer **C** is incorrect as this will send the print job to the default printer, which is not *EPSON_COLOR*. Answer **B** is incorrect as there is no `-sendmail` option and the `-C` option prints the class as the job classification on the banner page. Option **D** is incorrect as there is no `--sendmail` option.

4. A normal user wants to disable a printer in the CUPS Web interface. What do they need to do to achieve this?

   A. Enter the superuser user name and password when prompted by CUPS

   B. Pen a terminal window and enter the superuser name and password before launching the Web browser

   C. Start CUPS with the `-s` option

   D. Start CUPS and enter the superuser name and password in the **Authentication** tab

   Correct answers and explanations: **A**. Answer **A** is correct. CUPS will prompt the user for a username and password before a printer can be disabled. This is typically the *superuser password* or an administrative account with the appropriate privileges.

   Incorrect answers and explanations: **B**, **C**, and **D**. Answer **B** is incorrect as this will not pass the superuser credentials to the CUPS application. Answer **C** is incorrect as there is no `-s` option. Answer **D** is incorrect as there is not an authentication tab in the CUPS interface.

5. The main X Windows configuration file xinitrc is likely to be located in which system directory when GNOME has been installed as the only display manager?

A. */etc/X11*

B. */X1*

C. */usr/X11*

D. *User's home directory*

Correct answer and explanation: **D**. Answer **D** is correct as X Windows will always look in the user's home directory first and will use this configuration file if it is there.

Incorrect answers and explanations: **A**, **B**, and **C**. Answers **B** and **C** are incorrect as there are not any directories with that name defined as a standard. In addition, **A** is incorrect as the configuration file is not normally located there.

6. Which of the following would be the best description of the X Windows System if you were describing it to a new Linux user?

A. X Windows is a client-server architecture, with the client accepting keyboard input.

B. X Windows is a client-server architecture, with the server accepting keyboard input.

C. X Windows is a client-server architecture and cannot be ported to run on a Microsoft Windows system.

D. Both the X Windows server and client must be on the same system.

Correct answer and explanation: **B**. Answer **B** is correct as the X server accepts the input from the keyboard and mouse. In an X-server environment, the server provides display and I/O services to the applications.

Incorrect answers and explanations: **A**, **C**, and **D**. Answer **A** is incorrect as the clients in an X-server environment are the applications. Answer **C** is incorrect as X windows is platform independent. There are versions that can run on a multitude of operating system and can run on Microsoft Windows XP and Vista. Answer **D** is incorrect as the client and server may be on different systems.

7. A system administrator wants to add remote displays to systems configured with X Windows. Which protocol and port will be used between the X server and the remote client?

A. *XDMP* normally running on port 177

B. *XDMCP* normally running on port 177

C. *XDMP* normally running on port 187

D. *XDMCP* normally running on port 187

Correct answer and explanation: **B**. The correct answer is **B** as the protocol is XDMCP and the correct port 177.

Incorrect answers and explanations: **A**, **C**, and **D**. Answer **A** is incorrect as XDMP is not a valid protocol. Answer **C** is incorrect as XDMP is not a valid protocol and the port is incorrect. Answer **D** is incorrect as protocol XDMCP is correct but the port is incorrect.

8. A user has installed Linux on his/her system and has made KDE his/her default desktop manager. What is the default number of virtual desktops and the maximum number that can be configured by the user?

A. Default of four desktops and a maximum of 36

B. Default of two desktops and a maximum of 36

C. Default of two desktops and no maximum

D. Default of two desktops and the user cannot configure any more

Correct answer and explanation: **A**. Answer **A** is correct as KDE will normally default to four virtual desktops and the maximum allowable is 36.

Incorrect answers and explanations: **B**, **C**, and **D**. Answer **B** is partially correct in that the maximum is 36 desktops, but the default is four virtual desktops. Answer **C** is incorrect as the default is two and the maximum allowable is 36. Answer **D** is incorrect as there is a default of four virtual desktops and this can be increased to 36 by the user.

9. A company has installed Linux with the GNOME desktop on a number of older systems as a means to extend their life. These systems use an 800 × 600 display and have a maximum of 64 MB of RAM installed. They want to use these systems as remote terminal emulators to a more powerful X server on another system. Which remote terminal would give the best performance due to its small memory footprint?

A. `kconsole`

B. `gconsole`

C. `gnome-terminal`

D. `xterm`

Correct answer and explanation: **D**. Answer **D** is correct as `xterm` has a very small memory footprint (less than 1 MB) and is therefore likely to give the best overall performance.

Incorrect answers and explanations: **A**, **B**, and **C**. Answer **A** is incorrect as this is the normal `console` in KDE and GNOME is installed. Answer **C** is incorrect as the `gnome-terminal` has a much larger memory footprint than `xterm`. Answer **B** is incorrect as there is not a `gconsole` program as standard within GNOME.

10. A user is configuring his/her Linux system, which has KDE installed on it. The user wishes to add a new printer and will do so through the CUPS Web interface. The user has installed Firefox and the CUPS server on his/her system and both are working correctly. What would be the best way to access the CUPS server?

A. CUPS can only be accessed from the command line using the command `CUPS -S` when it is installed locally.

B. CUPS can be accessed using Firefox with the URL *http://localhost:631*.

C. CUPS can be accessed using Firefox with the URL *http://631:localhost*.

D. CUPS can be accessed using Firefox with the URL *http://cups@localhost*.

Correct answer and explanation: **B**. Answer **B** is correct as the CUPS server is hosted locally and the normal port it runs on is 631.

Incorrect answers and explanations: **A**, **C**, and **D**. Answer **A** is incorrect as CUPS is accessed normally from the Web interface. Answer **C** is incorrect as the URL is formatted incorrectly, the correct notation being *address:port*. Answer **D** is incorrect as no port is specified, and the URL looks more like an e-mail address.

11. A user has problems with the startup of his/her system and he/she wishes to start up his/her system up in single user mode and then to start X Windows. Which is the best method to achieve this?

A. Start the system in run level 1 and then run `startx`

B. Start the system in run level 5 and then run `startx`

C. Reboot the system and when the initial load screen appears, type **Ctrl** and **S** together

D. Reboot the system and when the initial load screen appears, type **Ctrl** and **1** together

Correct answer and explanation: **A**. Answer **A** is correct as this will start Linux in single user mode on most Linux distributions and will then allow the user to start X Windows when the command prompt appears.

Incorrect answers and explanations: **B**, **C**, and **D**. Answer **B** is incorrect as run level 5 normally starts X Windows at the end of the boot process. Answers **C** and **D** are incorrect as these commands are not available.

12. You have just downloaded and installed the latest version of the GNOME desktop. This is a beta version and your system seems to freeze when you start it. Which option would be the worst one to use?

A. Press the **reset** button and boot into single user mode with command- line input. Then remove the beta version.

B. Open a terminal window and type `shutdown -now` and then boot into single user mode with command- line input. Then, remove the beta version.

C. Type **Ctrl** + **Alt** + **F2** and use the root username and password when prompted. Look at the PID list and kill all the processes associated with X Terminal session. You can now uninstall the beta version.

D. Type **Ctrl** + **Alt** + **F7** and use the root username and password when prompted. Type `rollback X11` to revert to the previous version of X Windows.

Correct answer and explanation: **C**. Answer **C** is the correct answer. This will revert you to a terminal window where you can gracefully kill the rogue X Windows sessions and then remove the beta software.

Incorrect answers and explanations: **A**, **B**, and **D**. Answer **A** is incorrect as this will result in the loss of files that have not been written to the disk and possible disk corruption. It should only be used as a last resort. Answer **B** is also incorrect as this will also shut the system down immediately with the same consequences as **A**. Answer **D** is incorrect as there is not a `rollback` command to revert software back to a previous version.

13. You want to run the GNOME window manager if you boot your system into runlevel 5 and the Openbox window manager if you start the system in run level 4. How can this be best achieved?

A. You cannot start a different window manager based on runlevel.

B. Modify the .xinitrc in your home directory and include shell code to execute the commands `exec gnome-session` or `exec openbox-session` based on the current runlevel.

C. Boot the system directly into a terminal session and run a script to start X based on the appropriate runlevel.

D. Add the lines in the .xinitrc file in your home directory to switch window managers:

```
if runlevel=4 then
gnome-session else
openbox-session
end
```

Correct answer and explanation: **B**. Answer **B** is correct as you can test for the appropriate runlevel in this shell script and call the appropriate window manager using the `exec` command.

Incorrect answers and explanations: **A**, **C**, and **D**. Answer **A** is incorrect as you can start different window mangers. Answer **C** could be made to work, but this is not the best way to achieve the result as it is not automatic. Answer **D** is incorrect as the code is wrong and there is no `exec` command.

14. You are a system administrator for a large company and a user wants to purchase a new color printer for his administrative assistant to produce sales literature. This printer is not the usual printer you purchase. What would be your best advice to the user to ensure the new printer works with his Linux system?

A. Find out the make and models of the printers he is considering to purchase and check the www.cups.org Web site to see if the printer's drivers can be downloaded.

B. Look at the printer manufacturer's Web site to see if the printers are listed as "plug and play" devices and hence will work seamlessly.

C. Tell the user he can buy any printer that has the "CUPS Compatible" logo on the box.

D. Tell the user he can buy any HP printer as they are all compatible with Linux through downloads on HP's Web site.

Correct answer and explanation: **A**. Answer **A** is correct as you can download a large number of drivers from the CUPS Web site and easily check if the printer has a driver written for it.

Incorrect answers and explanations: **B**, **C**, and **D**. Answer **B** is incorrect as "plug and play" printers refer to their compatibility with Microsoft Windows and not to Linux. Answer **C** is incorrect as there is not a "CUPS Compatible" logo defined. Answer **D** is incorrect as a lot of HP printers have Linux drivers, but not all do.

15. You have sent a job to your default printer and have seen that there are a lot of jobs before it. As you need the printout in a hurry, which is the best option?

A. Using the CUPS GUI, find an idle printer and move your job to this printer.

B. Move your job to the top of the queue on your default printer using the CUPS GUI.

C. Login as superuser on the CUPS GUY. Pause all the jobs on the printer ahead of your job so your job will start next.

D. Resend your job to the printer using the option `-priority` on the `lpr` command, which will insert the job to the head of the queue.

Correct answer and explanation: **A**. Answer **A** is correct as this will allow you to print your job without affecting anyone else.

Incorrect answers and explanations: **B**, **C**, and **D**. Answer **B** is incorrect as there is no command to move your job to the top of the queue within CUPS. Answer **C** is incorrect as, like **B**, there is no `pause` command. Answer **D** is incorrect as you cannot force a job to the top of the queue using `lpr`.

CHAPTER 9: INSTALLING, CONFIGURING AS A SERVER

1. A DHCP server can be used to set up the following on a client

A. Fixed IP addresses and DNS zone data

B. Routing and leased IP addresses

C. Leased IP address and default printer IP address

D. E-mail address of the user

Correct answer and explanation: **B**. A DHCP server leases IP addresses to clients as they connect, and it can also set up some routing information on the client as well.

Incorrect answers and explanations: **A**, **C**, and **D**. Answer **A** is incorrect as a DHCP server can specify IP addresses but it does not give out DNS zone data although it can specify DNS servers. Answer **C** is incorrect as a DHCP server does not specify any information on

default printers. Answer **D** is incorrect as the DHCP server cannot specify an e-mail address of the user.

2. A DNS server has just been set up in your company. The primary purpose of this server is to

A. Enable Web browsing to occur

B. Translate domain names to IP addresses

C. Act as a gateway for users who wish to browse the Internet

D. Act as a file and print server for Microsoft Windows client

Correct answer and explanation: **B**. Answer **B** is correct as the primary purpose of the DNS server is to translate names meaningful to a human into actual IP addresses.

Incorrect answers and explanations: **A**, **C**, and **D**. Answer **A** is incorrect as although it does allow a user to put names into a browser, this is not the primary purpose. Answer **C** is incorrect as a DNS server cannot act as a gateway. This is performed by a proxy server. Answer **D** is incorrect as file and print services for Microsoft clients are provided by a Samba server.

3. You have just installed an NTP server onto your computer and want to set up a number of time servers in the configuration file. If you performed a standard installation, what file do you need to edit?

A. /etc/ntp.conf

B. /etc/ntpd.conf

C. /sys/ntpd.conf

D. /bin/ntp.conf

Correct answer and explanation: **A**. Answer **A** is correct as the file ntp.conf contains the NTP configuration data and is normally located in /etc.

Incorrect answers and explanations: **B**, **C**, and **D**. Answer **B** is incorrect as there is not an /etc/ntpd.conf file. Answer **C** is incorrect as there is not an /sys/ntpd.conf file. Answer **D** is incorrect as the ntp.conf file is normally located in the /etc directory.

4. You have just installed an Apache 2 Web server onto a server. The installation was successful and you now wish to start the server. What command would best accomplish this?

A. `apachectl start`

B. `apacheweb start`

C. `apache start`

D. `httpd -k start`

Correct answer and explanation: **A**. Answer **A** is correct as the apachectl script correctly starts the server with the right environment variables.

Incorrect answers and explanations: **B**, **C**, and **D**. Answer **B** is incorrect as there is not an apacheweb script. Answer **C** is incorrect as the `apache` command does not exist. Answer **D** is incorrect as although this command would start the server, it is not recommended to do so. The best method is **A**, which will call the `httpd` command correctly.

5. Your company has installed a MySQL server on your network. You have used traceroute to confirm that you have network connectivity to that server. Which port would your client program use to connect to the MySQL server?

A. TCP port 631

B. UDP port 3306

C. TCP port 3306

D. TCP port 631 and UDP port 3306

Correct answer and explanation: **C**. Answer **C** is correct as the default port to connect to a MySQL server is TCP 33063306.

Incorrect answers and explanations: **A**, **B**, and **D**. Answer **A** is incorrect as port 631 is associated with the CUPS management interface. Answer **B** is incorrect as MySQL needs TCP port 3306 and not UDP port 3306. Answer **D** is incorrect as port 631 is used by CUPS and MySQL does not connect on UDP port 3306.

6. You are administrating a Samba server on your network. You want each user to connect to his/her own home directory. What configuration changes would you need?

A. Add the line `valid users = %S` in the smb.conf file

B. Add the line `brosweable = yes` in the smb.conf file

C. Add the line `home = %S` in the smb.conf file

D. Add the line `path =` ~ in the smb.conf file

Correct answer and explanation: **A**. Answer **A** is correct as setting this will allow users to connect to their home directory \\servername\username. It will require other parameters to allow these directories to be writable as well.

Incorrect answers and explanations: **B**, **C**, and **D**. Answer **B** is incorrect as this sets the home directories to be browsable and does not connect the users to their own directory. Answer **C** is incorrect

as this is not a valid parameter. Answer **D** is incorrect as this sets the path name when shares are being defined.

7. You have just added a DHCP server onto your network to reduce the network administration tasks you have. Before you turn on DHCP on each of the clients, you want to test the connection. Which is the best command to achieve this?

A. `dhcpd`

B. `bootp`

C. `dhcpcd`

D. `pumpd`

Correct answer and explanation: **C**. Answer **C** is correct as this will request an IP address from the DHCP server.

Incorrect answers and explanations: **A**, **B**, and **D**. Answer **A** is incorrect as although it looks like it should be the `dhcp daemon` command, the right syntax is `dhcpcd`. Answer **B** is incorrect as this is usually run at boot time to allocate an IP address. It is often used with windowless devices. Answer **D** is incorrect, as this is not a valid command. The correct command should have been `pump`.

8. You want to install a proxy server in your network and have chosen the Squid proxy. You have a mixed network of Linux, Sun Solaris, and Microsoft XP clients. You want to keep the default Squid port but allow for migration of existing clients. What would be the best setting for the squid.conf file?

A. `http_port 80`

B. `http_port 3128 8080`

C. `http_port 3128`

D. `http_port 3128 80`

Correct answer and explanation: **D**. Answer **D** is correct as this will keep the default Squid port (3128) and allow the existing Microsoft clients to still access the proxy through port 80, which is the default for Microsoft. You can systematically change this port to 3128 at your leisure.

Incorrect answers and explanations: **A**, **B**, and **C**. Answer **A** is incorrect as although this will allow most clients (particularly Microsoft ones) to access the proxy, it does not include the default Squid proxy port. Answer **B** is incorrect as this includes the default Squid proxy port but the normal proxy port for Microsoft is 80. Answer **C** is incorrect as this is the standard port for Squid,

but Microsoft clients would normally use port 80, which is not included.

9. In your DHCP server, you wish to allocate a fixed IP address to one of your color laser printers. What do you need to do to set this up?

A. Find the IP address the printer is currently assigned and fix it using the `dhcpcd fix IPaddress` command on your DHCP server.

B. Ensure bootp is available on the printer and then assign an IP address to this in the dhcpd.conf file with the host parameter.

C. Find the printers' MAC address and a spare IP address from the pool of IP addresses defined on your DHCP server and set this up in the dhcpd.conf file with the host parameter.

D. Set a fixed IP address directly on the printer and allow this to broadcast it to the DHCP server upon the printer being started.

Correct answer and explanation: **C**. Answer **C** is correct as you can set a fixed IP address against a specific MAC address using the host option.

Incorrect answers and explanations: **A**, **B**, and **D**. Answer **A** is incorrect as there is not a command option to `dhcpd` to fix the IP address. Answer **B** is incorrect as bootp is a program that will ask the DHCP server for an IP address. The DHCP server needs to know this is a fixed IP address and this has to be set up with the host option. Answer **D** is incorrect as this may cause an IP address conflict as the DHCP server has not allocated a fixed IP address to this and may issue that IP address to another client.

10. Your new Apache Web server has been set up and one of the developers wants to know which directory to load the Web pages. How can you find out which directory this is?

A. Look in the httpd.conf file for a `WebRoot` directive

B. Look in the httpd.conf file for a `DocumentRoot` directive

C. Look in the httpd.conf file for a `ServerName` directive

D. Look in the httpd.conf file for a `WebBase` directive

Correct answer and explanation: **B**. Answer **B** is correct as the DocumentRoot parameter correctly defines the root directory where the Web pages are stored.

Incorrect answers and explanations: **A**, **C**, and **D**. Answer **A** is incorrect as there is no WebRoot parameter. Answer **C** is incorrect as the ServerName parameter defines the name of the Web site. Answer **D** is incorrect because, again, there is no WebBase directive.

11. You have been told that your mail MTA is being used as a relay by spammers. You want to stop this happening. What is the best course of action?

A. Relocate the mailserver behind your corporate firewall and only allow TCP port 25 to and from this server.

B. Ensure that the only hosts that the mailserver will allow to relay are on your local network by configuring the */etc/mail/access* file.

C. Ensure that only your domain can be relayed by configuring the *relay-domains* file.

D. Configure the mailserver to stop all relaying of mail and make sure all the users connect to it via an approved client.

Correct answer and explanation: **B**. Answer **B** is correct as this will specify which of your networks can connect to this mailserver and have their mail relayed. If you configure the networks correctly, the spam mail will not be relayed.

Incorrect answers and explanations: **A**, **C**, and **D**. Answer **A** is incorrect as relocating the mailserver behind a firewall will not help. The server will have to have port 25 open on it, and this is used by spammers to force the mailserver to relay messages. This option would be useful if the correct configuration files as specified in Answer **A** were done. Answer **C** is incorrect as this will allow the domain users to still send e-mail, but spammers can easily spoof the domain name and could continue to use this as a relay mailserver. Answer **D** is incorrect because if you stop the mailserver from relaying all mail, users will not be able to send any mail.

12. The speed of your Internet connection has slowed down because of the increase in the number of employees in your company. You have installed a Squid proxy and now wish to restrict the browsing of certain sites to lunchtimes only. You have set the list of banned sites up as an acl called banned_sites. Which is the correct configuration in the squid.conf file?

A.
```
acl lunchtime MTWTFSS 12:00 13:00
http_access allow banned_sites lunchtime
```

B.
```
acl lunchtime MTWTFSS 12:00 13:00
http_access allow banned_sites lunchtime
http_access deny banned_sites NOT lunchtime
```

C.
```
acl lunchtime MTWHFAS 12:00 13:00
http_access allow banned_sites lunchtime
http_access deny banned_sites
```

D.
```
acl lunchtime MTWTFSS 12:00 13:00
http_access allow banned_sites lunchtime
http_deny banned_sites
```

Correct answer and explanation: **C**. Answer **C** is correct as this specifies the configuration parameters correctly. In particular, the days of the week must be specified in that form (with Thursday as H and Saturday as A).

Incorrect answers and explanations: **A**, **B**, and **D**. Answer **A** is incorrect as this specifies the days of the week incorrectly and also does not have a deny filter. Answer **B** is incorrect as this specifies the days of the week incorrectly and also the deny parameters are incorrect as NOT is invalid. Answer **D** is incorrect as there is no http_deny configuration parameter, the correct terminology is *http_access deny*.

13. You want to set up your Apache server to capture logs as you are having problem with the application. What log level would you set to give you the most verbose logs?

A. `emerg`

B. `error`

C. `info`

D. `alert`

Correct answer and explanation: **C**. Answer **C** is correct as this is the lowest log level and will give the most verbose output from the options listed. Debug can also be used, which will give a more verbose output than info.

Incorrect answers and explanations: **A**, **B**, and **D**. Answer **A** is incorrect as this will only give out emergency log message such as *Child cannot open lock file. Exiting.* Answer **B** is incorrect as this just gives logs of errors. Although this may be enough for your debugging, it does not give the most verbose logs. Answer **D** is incorrect as this will just give log messages for actions that must be taken.

14. The Samba server in your office has been set up with the name *samserv*. You want to connect to the sammy directory that has been set up and shared on it. What will be the correct command from a terminal shell if you want to connect as a user called juliet?

A. `smbclient //samserv/sammy juliet`

B. `smbclient //samserv/sammy -u juliet`

C. `smbclient //samserv/sammy -U juliet`

D. `smbclient //samserv/sammy U juliet`

Correct answer and explanation: **C**. Answer **C** is correct as this will invoke the smclient and connect to the share sammy on the target server (samserv). The response from the samba server should be *Password*: prompting the user to enter the password they have set up on that host.

Incorrect answers and explanations: **A**, **B**, and **D**. Answer **A** is incorrect as the option to specify the user (`-U`) is missing. Answer **B** is incorrect as the option to specify a user name is `-U` and `-u`. Answer **D** is incorrect as this does not specify the user option correctly.

15. You want to administer your DNS server using the `rndc` command, but you cannot connect to the server. You have pinged the server and it responds. You have just installed the client on your machine. What is the likely error?
    - **A.** You have not put your machines' IP address into the rndc.conf file for the target DNS.
    - **B.** You have not inserted the correct keys into the rdnc.conf file.
    - **C.** You have run the `dnssec-keygen` command immediately before issuing the `rndc` command.
    - **D.** You must run the `dnssync` command on the new host and the target to ensure they can communicate with each other.

    Correct answer and explanation: **B**. The correct answer is **B** because the `rdnc` command needs to have authentication keys that match those of the target server.

    Incorrect answers and explanations: **A**, **C**, and **D**. Answer **A** is incorrect as you do not need to insert the IP address of the target DNS in that file. Answer **C** is incorrect because although you need to use this command to get the correct authentication key, the value must be put into the rndc.cong file. Answer **D** is incorrect as there is no command called `dnssync`.

CHAPTER 10: SECURING LINUX

1. HR calls to tell you Susie Smith got married and needs her Linux user account changed from ssmith to sjenkins. What command will change Susie's user account and change her home directory?
    - **A.** `useradd -c sjenkins smith`
    - **B.** `usermod -c sjenkins -d /home/sjenkins -m ssmith`
    - **C.** `umod sjenkins ssmith`
    - **D.** `uname sjenkins ssmith`

Correct answer and explanation: **B**. **B** is the correct answer because `usermod` is the command to change user information, including uid, gid, username, and home directory among others. Note that the `-m` is needed to actually move Susie's files.

Incorrect answers and explanations: **A**, **C**, and **D**. **A** is incorrect because `useradd` is used to create new accounts, not to modify existing ones. Although it is possible to delete Susie's old account and then create a new one, sorting out the ownership of her existing files would have to be dealt with. **C** is incorrect because `umod` is not a valid command. **D** is incorrect because `uname` gives information about the Linux system, such as the kernel version and processor type.

2. Bob just came back from vacation, where he had such a good time forgetting about work that now he can't remember the password to his Linux account. Which of the following commands could you use to reset it for him if you are logged in as root?

A. `sudo -u bob passwd`
B. `passwd bob`
C. usermod -p <new_password> bob
D. `password bob`

Correct answer and explanation: **B**. **B** is the correct answer because the `passwd` command is used to change a user's password if the user account is provided. If no user account is given, it will change the password of the current logged-in user.

Incorrect answers and explanations: **A**, **C**, and **D**. **A** is incorrect because `sudo -u bob passwd` will run the `passwd` command as user bob but requires bob's current password before allowing a new one to be entered. **C** is incorrect because `usermod -p`, although it WILL change his password, assumes that the password entered is in encrypted form, not plain text. **D** is incorrect because `password` isn't a valid Linux command – recall that `passwd` is the correct form.

3. Your boss wants you to find out how many devices are on your subnet right now because he is thinking of adding another 20 machines and doesn't want to run out of addresses. What's a good way you could check?

A. Use `wireshark` to monitor network traffic
B. Use `snort` to scan the network
C. Use `tripwire` to scan the network
D. Use `nmap` to scan the network

Correct answer and explanation: **D**. **D** is the correct answer because `nmap` can be used to scan a subnet and list devices connected to the network.

Incorrect answers and explanations: **A**, **B**, and **C**. **A** is incorrect because `wireshark` only monitors traffic that makes it to your network interface and may not see devices that are idle. **B** is incorrect because `snort` monitors traffic seen by your local network interface and typically only pays attention to traffic that matches a list of known suspicious patterns. **C** is incorrect because `tripwire` monitors files on the local host for potentially unauthorized changes.

4. You just finished downloading the latest version of your favorite program and want to make sure that it downloaded correctly by comparing it with the hash file posted on the download site. What program would you use to make sure the file is exactly what it should be?

A. `filecheck my_download`

B. `md5sum -c my_download.md5`

C. `diff my_download.md5`

D. `gpg my_download`

Correct answer and explanation: **B**. **B** is correct because `md5sum -c` will calculate the hash value of the file listed in the given file and compare it with the hash value in the file.

Incorrect answers and explanations: **A**, **C**, and **D**. **A** is incorrect because `filecheck` isn't a valid Linux command. **C** is incorrect because the `diff` command is used to compare two local files. **D** is incorrect because `gpg` would be used to confirm the file is correct if it has been signed by the owners' private key using your copy of his public key.

5. You are going to start working from home occasionally and need full access to your office Linux workstation as well as a windows PC via your VPN connection. What remote access software should you consider?

A. ssh

B. X11

C. VNC

D. PCAnywhere

Correct answer and explanation: **C**. **C** is the correct answer because VNC works between unlike operating systems and provides full desktop access.

Incorrect answers and explanations: **A**, **B**, and **D**. **A** is incorrect because ssh doesn't give access to windows desktops. **B** is incorrect because X11 isn't supported on Windows. **D** is incorrect because PCAnywhere isn't supported on Linux.

6. Your company is starting a new project, and you need to create a shared directory for the project team members to share their documents. You want all the new documents created in the directory to be automatically set to allow their owner and other group members to read and write them and all other system users to read only. What commands could you use?

A. `file -default ug+rw, o+r`

B. `chmod -default 664`

C. `umask 0022`

D. `umask 0002`

Correct answer and explanation: **D**. **D** is the correct answer because `umask 0002` will set the mode mask for the directory to 0002, which will set the default mode for new files to 664, which is read/write for users and group members and read-only for everyone else.

Incorrect answers and explanations: **A**, **B**, and **C**. **A** is incorrect because the `file` command gives information about types of files but doesn't change any file modes. **B** is incorrect because `chmod` changes the mode of specific files but doesn't change the default mask used during the creation of new files. **C** is incorrect because 0022 will set the mask for owner read/write, group, and other to read only.

7. You've done such a great job of showing how cool Linux is that now there are a bunch of new Linux machines. To make your job easier, you want to consolidate user information in one place, instead of having to make user accounts on each machine. What system, or systems, could you implement to centralize user information?

A. shadow

B. NIS

C. RADIUS

D. LDAP

Correct answers and explanations: **B** and **D**. **B** is correct because NIS (Network Information Systems) supports centralizing user and

system configuration information. **D** is correct because LDAP also allows user information to be centralized on a single server and accessed on client workstations.

Incorrect answers and explanations: **A** and **C**. **A** is incorrect because /etc/shadow is used to store encrypted account password information locally on a Linux system, but it isn't centralized. **C** is incorrect because RADIUS allows centralized authentication but doesn't support other account information.

8. Bob has been assigned to help out the testing_development project and needs to be added to the appropriate group so he can get access to the groups shared files. What command will add Bob to the correct group?

A. `usermod -g testing_dev`

B. `groupmod -u bob`

C. `usermod -G testing_dev`

D. `usermod -Ga testing_dev`

Correct answer and explanation: **D**. **D** is the correct answer because `usermod -Ga testing_dev` will add bob to the testing_dev group.

Incorrect answers and explanations: **A**, **B**, and **C**. **A** is incorrect because `usermod -g` is used to change the users' primary group, not to add him to an additional group. **B** is incorrect because `groupmod` doesn't offer an option for adding users to a group. **C** is incorrect because `usermod -G` without the `-a` option will replace any other groups bob may be a member of, instead of adding him to the testing_dev group.

9. You need to take your Linux system offline for maintenance and want to check to see who else may be using it, so you can be courteous and let your users know about it. What command, or commands, will show who else is logged into your system?

A. `w`

B. `finger`

C. `who`

D. `lsof`

Correct answers and explanations: **A**, **B**, and **C**. **A** is correct because the `w` command will show who is currently logged in, what programs they are running, and where they are connected from. **B** is correct because `finger` will also show who is connected to a

Linux machine, although it is frequently disabled for security reasons. **C** is correct because `who` will show who is connected to a Linux machine, too.

Incorrect answer and explanation: **D**. **D** is incorrect because `lsof` is used to give a listing of all open files, including files open on the system, including local system files.

10. Your network seems to be running slower than normal, and many users are complaining of odd things happening on their computers. You suspect your company may be the victim of the latest computer virus. What tool could you use to check?

A. `ssh`

B. `nessus`

C. `SNORT`

D. `tripwire`

Correct answer and explanation: **C**. **C** is the correct answer because SNORT is a program that monitors network traffic for suspicious traffic, including viruses.

Incorrect answers and explanations: **A**, **B**, and **D**. **A** is incorrect because ssh is used to encrypt traffic between computer systems. **B** is incorrect because Nessus is used to scan for vulnerabilities but won't tell you whether a given vulnerability has been exploited by an active virus. **D** is incorrect because tripwire is used to monitor files for unauthorized changes. If you have it installed and configured beforehand, it could show virus activity but wouldn't be able to detect a preexisting virus.

11. You need to create a new group to support a new product roll-out. What command, or commands, will let you make a new account?

A. `addgroup project_x`

B. `groupadd -g project_x`

C. `newgroup project_x`

D. `groupadd project_x`

Correct answers and explanations: **A** and **D**. **A** is correct because the `addgroup` command is used to create a new groups on some distributions. **D** is correct because `groupadd` is used to create a new group on other distributions. It is more generally supported than `addgroup` but doesn't have as many options.

Incorrect answers and explanations: **B** and **C**. **B** is incorrect because the `-g` option is used to specify a numeric GID, not an

alphanumeric name. **C** is incorrect because `newgroup` isn't a valid Linux command.

12. You want to tighten security on a particular Linux computer by limiting which users have access to the `sudo` command. Which file should you edit to lock down this feature?

A. /etc/users

B. /etc/shadow

C. /etc/sudoers

D. /etc/passwd

Correct answer and explanation: **C**. **C** is `sudo` security that is controlled by the /etc/sudoers file.

Incorrect answers and explanations: **A**, **B**, and **D**. **A** is incorrect because /etc/users isn't a standard Linux file. **B** is incorrect because /etc/shadow is used to store encrypted user passwords. **D** is incorrect because /etc/passwd is used to store user account information but not `sudo` access rights.

13. You need to set a shared file for read and write access for the file owner and members of the files group and no access for anyone else. Which command(s) will give the desired result?

A. `chmod 440 shared_file`

B. `chmod 660 shared_file`

C. `chmod ug=rw,o=`

D. `chmod og=r,e=`

Correct answers and explanations: **B** and **C**. **B** is correct because the octal mode bits 660 equate to read (4) and write (2), totaling six for both user (first position) and group (second position), with no rights (0) for everyone else (third position). **C** is correct because it sets rights for user (u) and group (g) to read (r) and write (w), and sets everyone else (o) to nothing by leaving the field blank. Note that the different groups of users need to be separated by a comma.

Incorrect answers and explanations: **A** and **D**. **A** is incorrect because octal mode 440 would set owner and group rights to read only. **D** is incorrect because the file owner is correctly abbreviated with a "u" (think user) and everyone else is represented by an "o" for "other," not an "e."

14. You are testing out SELinux to enhance security on your Linux computer. What mode would you use to let all programs run, but log anything that would fail if you were to lock it down?

A. *enabled*

B. *allowed*

C. *permissive*

D. *test*

Correct answer and explanation: **C**. **C** is the correct answer because in permissive mode SELinux is engaged but doesn't block programs, only logs what it would block if it were set to "*enabled.*"

Incorrect answers and explanations: **A**, **B**, and **D**. **A** is incorrect because in enabled mode SELinux prevents programs that violate defined policies from running. **B** and **D** are incorrect because allowed and test modes aren't valid SELinux running modes.

15. You are running out of room on your backup system and want to flag a large temporary file so the tape backup system skips it. What is a way you could do that?

A. `chmod -s temp_file`

B. `setattr -d temp_file`

C. `chattr -d temp_file`

D. `attr -d temp_file`

Correct answer and explanation: **C**. **C** is the correct answer because the `chattr` command is used to set file attributes in Linux and the `-d` option is used to flag a file to be skipped by backup systems (`d` references *dump*, a basic backup program).

Incorrect answers and explanations: **A**, **B**, and **D**. **A** is incorrect because `chmod` is used to change file permissions, not attributes. **B** and **D** are incorrect because `setattr` and `attr` are not valid Linux commands.

CHAPTER 11: TROUBLESHOOTING AND MAINTAINING LINUX

1. Which of the following commands does not display load average?

A. `top`

B. `w`

C. `who`

D. `uptime`

Correct answer and explanation: **C**. Answer **C** is correct because the `who` command displays the usernames of those who are logged into the system.

Incorrect answers and explanations: **A**, **B**, and **D**. Answers **A**, **B**, and **D** are incorrect because all of them, and several other commands, display the load average.

2. What command would you use to generate a static report of CPU utilization?

A. `uptime`

B. `vmstat`

C. `top`

D. `iostat`

Correct answer and explanation: **D**. Answer **D** is correct because `iostat` is the only command among the available answers that will return a point-in-time report of CPU utilization.

Incorrect answers and explanations: **A**, **B**, and **C**. Answer **A** is incorrect because uptime will not return CPU utilization statistics. Answer **B** is incorrect because `vmstat` delivers statistics on virtual memory utilization. Answer **C** is incorrect because, while `top` will return CPU utilization statistics, its output is not static; it provides real-time reporting.

3. You want to brag about how long it has been since your server was last rebooted to your colleagues who manage servers that run a different operating system. What is the best command to use to find out how long it has been since your last reboot?

A. `sar`

B. `uptime`

C. `iostat`

D. `loadav`

Correct answer and explanation: **B**. Answer **B** is correct because `uptime` is the simplest and easiest way to display how long your server has been up and running.

Incorrect answers and explanations: **A**, **C**, and **D**. Answer **A** is incorrect because `sar` will only return the statistics captured at a given interval, not the overall uptime. Answer **C** is incorrect because `iostat` returns CPU and hard disk utilization and performance statistics. Answer **D** is incorrect because there is no command called `loadav`.

4. In the following list, what is not a valid *syslog* event level?

A. Emerg

B. Alarm

C. Err

D. Notice

Correct answer and explanation: **B**. Answer **B** is correct because Alarm is not a valid *syslog* event level. There is a valid event level entitled, Alert.

Incorrect answers and explanations: **A**, **C**, and **D**. Answers **A**, **C**, and **D** are incorrect because they are all valid *syslog* event levels.

5. The performance of your corporate SMTP relay server has been intermittently slow and you suspect a hardware problem. Which of the following log files would you use to look for hardware-related events?

A. /var/log/messages

B. /var/log/syslog

C. /var/log/maillog

D. /var/log/secure

Correct answer and explanation: **A**. Answer **A** is correct because /var/log/messages capture hardware events.

Incorrect answers and explanations: **B**, **C**, and **D**. Answer **B** is incorrect because /var/log/syslog is used to collect information about authorization events, time changes, and system statistics events. Answer **C** is incorrect because /var/log/maillog is used to collect information about mail server (software, for example, sendmail) events. Answer **D** is incorrect because /var/log/secure is used to collect information about sudo, sshd, and other security events on Redhat-based distributions only.

6. You are trying to get a user's USB flash drive to mount on a Linux workstation and are experiencing trouble. As part of your troubleshooting, you decide that you want to find out if the system is recognizing that the flash drive has been inserted. Using the output of the `dmesg` command as the source, what is the correct syntax for *grep* to find all USB-related events?

A. `dmesg | grep ‘usb’`

B. `dmesg | grep “USB”`

C. `dmesg | grep -i ‘usb’`

D. `dmesg | grep -i usb`

Correct answer and explanation: **D**. Answer **D** is correct because it will return all USB-related events, regardless of whether the string "USB" is capitalized or not.

Incorrect answers and explanations: **A**, **B**, and **C**. Answer **A** is incorrect because the search string in *grep* needs to be in double quotation marks. Answer **B** is incorrect because, although the command will return all events that contain the string "USB," it will not return any events that contain "usb," which may hamper troubleshooting. Answer **C** is incorrect because even though it will perform a case-insensitive search, the search string must be in double quotation marks.

7. A user is using FTP to upload a graphics file to a remote server and when the file is loaded in a browser, the browser window is filled with gibberish. What should the user be doing to prevent this from happening?

A. type `bin` at the FTP prompt to transfer files in binary mode

B. type `ascii` at the FTP prompt to transfer files in ASCII mode

C. type `pasv` to force the FTP connection into passive mode

D. use `mput` instead of `put` to transfer the file

Correct answer and explanation: **A**. Answer **A** is correct because FTP by default is in ASCII mode and the graphic file is being converted to text. Binary mode will ensure that the graphic file is transferred and stored in its native format.

Incorrect answers and explanations: **B**, **C**, and **D**. Answer **B** is incorrect because the FTP server is already in ACSII mode and a graphic should be transferred using binary mode. Answer **C** is incorrect because the problem does not lie in whether or not the FTP server is in passive or active mode; the problem is with the transfer method. Answer **D** is incorrect because the problem is not with the command used to transfer the file; `mput` is used to transfer multiple files.

8. Users in your finance department are reporting errors when trying to connect to their server. You decide to monitor activity on this server as the users try to connect. What command would you use with `dmesg` to monitor these system events?

A. `dmesg | tail -f`

B. `dmesg | less`

C. `dmesg | tail -n5`

D. `dmesg | less -h5`

Correct answer and explanation: **A**. Answer **A** is correct because the `tail` command the `-f` switch will result in the output of `dmesg` being displayed to the screen as events are captured. An easy way to

remember this switch is that you would use `-f` to "follow" system events as they happen.

Incorrect answers and explanations: **B**, **C**, and **D**. Answer **B** is incorrect because while `less` will print the last (and newest) portion of the log, the command will need to be run over and over again; `tail -f` will show events as they are captured automatically. Answer **C** is incorrect because `tail` will only display the last five lines from the `dmesg` output once, not dynamically as users are connecting. Answer **D** is incorrect because `less` will only display the last five lines from the `dmesg` output once, not dynamically as users are connecting.

9. You have been asked to create a CD that contains the personal files of a user who is leaving your company. The files are stored on your Linux-based file server in /home/miranda and the CD needs to be readable on a Windows computer. What is the correct syntax to create the image file for the CD?

A. `mkisofs -rW -o mirandasfiles.iso /home/miranda`

B. `mkisofs -rJ -o mirandasfiles.iso /home/miranda`

C. `mkisofs -r -o mirandasfiles.iso /home/miranda`

D. `cdrecord -rJ -o mirandasfiles.iso /home/miranda`

Correct answer and explanation: **B**. Answer **B** is correct because the proper command is `mkisofs` and the switch needed to ensure readability on a Windows-based computer is `-J`.

Incorrect answers and explanations: **A**, **C**, and **D**. Answer **A** is incorrect because `-W` is an invalid switch; the required switch is `-J`. Answer **C** is incorrect because the `-J` switch is missing. Answer **D** is incorrect because `mkisofs` is the required command; `cdrecord` is used to burn the image.

10. You are managing Web servers in both a development environment and on the Internet (hostname is www). Once development on a given release is complete and tested, the developers ask you for a solution to keep the content on the staging server in the development environment synchronized with the content on the public Web server. What command would you run on the staging server to ensure that the files and all subdirectories on the staging server are updated on the public Web server as they are updated?

A. `rsync * www:/home/httpdocs`

B. `ftp www | put *`

C. `rsync -r * www:/home/httpdocs`

D. `ssh www | copy -r * www:/home/httpdocs`

Correct answer and explanation: **C**. Answer **C** is correct because `rsync` is the proper command to only transfer files that have changed, and the `-r` switch is needed to ensure that subdirectories are included in the transfer.

Incorrect answers and explanations: **A**, **B**, and **D**. Answer **A** is incorrect because it is missing the `-r` switch. Answer **B** is incorrect because, while `ftp` is used to transfer files, it will not compare files to see if they have changed; furthermore, one cannot pipe `ftp` commands because it opens its own shell. Answer **D** is incorrect because the syntax for ssh is incorrect and `copy` will not compare files to see whether they have changed.

11. You have been asked to back up users' data on a particular server before a core application is upgraded. Because of the amount of data, you need to ensure that these files will fit on a remote hard disk. What command would you use to ensure the smallest possible size of the backup file?

A. `tar -cvf userdata.tar /home/*`

B. `tar -xjvf userdata.tar /home/*`

C. `tar -cjvf userdata.tar /home/*`

D. `tar -xvf userdata.tar /home/*`

Correct answer and explanation: **C**. Answer **C** is correct because the command with the `-c` and `-j` switches create the archive and compress the files in the archive with bzip, respectively.

Incorrect answers and explanations: **A**, **B**, and **D**. Answer **A** is incorrect because, while the `-c` switch creates the tarfile, there is nothing to instruct `tar` to compress the files in the archive. Answer **B** is incorrect because the `-x` switch is used to extract the files from the archive, which in this case contains compressed files. Answer **D** is incorrect because the `-x` switch is used to extract the files from the archive.

12. You are replacing Michael's computer and have backed up his hard disk to an attached USB external hard disk (/mount/usbhdd) using the following syntax: `dump -0uf- A michaelhdd.archive -f /mount/usbhdd/michaelhdd.backup /`. You want to restore the backup on another hard disk in the new computer. After booting

the new computer and mounting the external hard disk, what command do you use?

A. `restore -rf /mount/usbhdd/michaelhdd.backup`

B. `dump -xf /mount/usbhdd/michaelhdd.backup`

C. `tar -xvf /mount/usbhdd/michaelhdd.backup`

D. `restore -rf /mount/usbhdd/michaelhdd.archive`

Correct answer and explanation: **D**. Answer **D** is correct because the `restore` command with the `-rf` switches is required to extract all files from the archive and specify the archive file.

Incorrect answers and explanations: **A**, **B**, and **C**. Answer **A** is incorrect because the output file is specified, and the `f` switch is used to specify the archive file. Answer **B** is incorrect because `dump` is used to back up the files and has no restore functionality. Answer **C** is incorrect because only `restore` can be used to extract files from a backup set created with `dump`.

13. Lately you have been hearing reports that your Linux server is slow to respond and you have a suspicion that there are applications that are consuming more than their fair share of the server's memory. What key combination would you press while `top` is running so that the running programs are sorted by their respective percentage of memory utilization?

A. *F*, then *M*

B. *F*, then *n*

C. *F*, then *k*

D. *F*, then *l*

Correct answer and explanation: **B**. Answer **B** is correct because *n* will display the percentage of memory usage.

Incorrect answers and explanations: **A**, **C**, and **D**. Answer **A** is incorrect because *M* will display the CPU time in hundredths of a second. Answer **C** is incorrect because *k* will sort by percentage of CPU utilization. Answer **D** is incorrect because *l* displays utilization based on CPU time.

14. Users are reporting that a particular corporate server responds slowly for around 30 min between 10:30 A.M. and 11:00 A.M. You decide to run *sar* at regular intervals during this time to capture statistics on the server's network performance. What syntax would you use to capture six sets of these metrics every 10 min?

A. `sar -A DEV 600 6`

B. `sar -n DEV 600 6`

C. `sar -n DEV 6 600`

D. `sar -A DEV 6 600`

Correct answer and explanation: **B**. Answer **B** is correct because the switch and parameters needed to collect the desired statistics `-n DEV 600 6`. The `-n` switch tells *sar* to capture network statistics; the next parameter, 600, specifies the interval in seconds; and the last interval, 6, specifies that six sets of statistics should be captured.

Incorrect answers and explanations: **A**, **C**, and **D**. Answer **A** is incorrect because `-A` tells *sar* to capture all available system statistics. Answer **C** is incorrect because the last two parameters are inverted, which means that 600 sets of statistics will be captured at six-second intervals. Answer **D** is incorrect because the `-A` switch is used and the parameters are inverted.

15. You are in the process of setting up an active mode FTP server. Whenever you try to connect, you can connect to the server, but you cannot enter a username and password. You made sure that TCP ports 21 and 20 are open on the server. What is the most probable cause of the problem?

A. TCP ports 1022 and below are open on the server

B. TCP ports 1023 and above are open on the server

C. TCP ports 1022 and below are closed on the server

D. TCP ports 1023 and above are closed on the server

Correct answer and explanation: **D**. Answer **D** is correct because TCP ports 1023 and above need to be open to ensure that all the connections required for active mode can be established.

Incorrect answers and explanations: **A**, **B**, and **C**. Answers **A** and **C** are incorrect because active mode FTP uses TCP ports 1023 and above are required, not below. Answer **B** is incorrect because the connection could be established if the ports were open.

Glossary

The following terms show up throughout the book and, with their accompanying definitions, they should be a useful resource when studying for the exam.

Advanced Configuration and Power Interface (ACPI) A specification that establishes common interfaces between an operating system (OS)-configured motherboard and device-based power management.

Advanced Packaging Tool (APT) A front end to the core package management system on Debian-based Linux distributions (dpkg) to install, manage, and remove software packages.

American Standard Code for Information Interchange (ASCII) A character-encoding scheme based on the ordering of the English alphabet. ASCII codes are the numerical representation of a test character in computers, communications equipment, and other devices that work with text.

Apache An open-source HTTP (Web) server produced by the Apache Software Foundation that has become the most widely used Web server on the Internet. It aims to be aligned with current HTTP standards and to run on all modern operating systems.

Berkeley Internet Naming Daemon (BIND) An open-source server that implements the DNS protocols for the Internet. BIND is by far the most widely used DNS software on the Internet.

Bourne Again Shell (BASH) The sh-compatible command line interpreter that executes commands read from the standard input or from a file. It is the default shell on most Linux distributions.

Command line interpreter or command line interface (CLI) A full-screen or windowed text-mode session where the user executes programs by typing in commands with or without parameters. The CLI displays output text from the operating system or program and provides a command prompt for user input.

Common Gateway Interface (CGI) A standard for interfacing external applications with information servers, such as HTTP or Web servers. A CGI program is executed in real time so that it can output dynamic information.

Common Internet File System (CIFS) An interoperable mechanism, developed by Microsoft, for a client system to request file access from a server system in a network, regardless of the underlying operating system platforms of the respective systems.

Common UNIX Printing System (CUPS) The standard-based, open-source printing system developed by Apple Inc. for Mac OS X and other UNIX-like operating systems.

Compact disc (CD) A 4.72-in. disc developed by Sony and Philips that can store, on the same disc, still and/or moving images in monochrome and/or color; stereo or two separate sound tracks integrated with and/or separate from the images; and digital program and information files.

Deb The file format used for packaging applications for installation using dpkg.

Digital versatile disc (DVD) An optical storage medium with improved capacity and bandwidth compared to a CD. DVD, like CD, was initially marketed for entertainment and later for computer users. A DVD can store 4.7 GB, the equivalent of a full-length film with up to 133 min of high-quality video, in MPEG-2 format, and audio. In addition, the DVD-ROM drive can read DVD movies, and modern computers with the appropriate hardware or software can decode them in real time.

Discretionary Access Control (DAC) A type of access control where system privileges are specified by the owner of an object, who can apply, modify, or remove them at will.

Domain name service (DNS) A hierarchical naming system for computers, services, or any resource connected to the Internet. It associates information, such as IP addresses, aliases, and resource types, with domain names assigned to each device or service. It's most important task is to resolve IP addresses with domain and host names and vice versa.

dpkg The core package management system on Debian-based Linux distributions.

Dynamic Host Configuration Protocol (DHCP) A protocol for automating the configuration of computers that use TCP/IP. DHCP can be used to automatically assign IP addresses, to deliver TCP/IP stack configuration parameters such as the subnet mask and default router, and to provide other configuration information such as the addresses for printer, time, and news servers.

Environment variables A set of dynamic named values that can affect the way running processes will behave on a computer. In Linux (and UNIX), they are initialized on startup through init scripts and inherited by all other running processes.

EXT2 (second extended filesystem) The default filesystem in several Linux distributions until EXT3 was released.

EXT3 A journaled filesystem is commonly used by the Linux kernel. It is almost completely backwards compatible with EXT2.

File Allocation Table (FAT) A filesystem named after the table that an operating system maintains on a hard disk that provides a map of the clusters in which a file has been stored.

File Transfer Protocol (FTP) A client–server protocol that allows a user on one computer to transfer files to and from another computer over a TCP/IP network. Also the user executes the client program to transfer files. An FTP server in passive mode uses TCP port 21 for the command port and an unprivileged port numbered 1023 or higher; active mode uses TCP ports 20 (data port) and 21 (command port) for transmission.

Filesystem A filesystem provides the operating system with a framework (a structure) for the storage, organization, modification, removal, and retrieval of digital information. It is responsible for organizing and maintaining files, folders, metadata, and residual data as containers for storing digital information.

Gigabyte (GB) A measure of computer data storage where 1 GB is equal to 1,073,741,824 bytes, or 1073 MB.

GNOME Display Manager (GDM) The open-source graphical login program from the Gnome Project. Like KDE, it is the Gnome replacement for XDM, the default X display manager, that runs in the background to run X11 sessions and enables a graphical login.

GNU (GNU is not UNIX) A project that was launched in 1984 to develop the GNU system, a complete UNIX-like operating system comprised of free software. It is the author of the most widely used free software licenses, such as the GNU General Public License (GPL).

GNU Privacy Guard (GPG) An implementation of the OpenPGP standard that allows you to encrypt and sign your data and communication. It features a versatile key management system and access modules for all kind of public key directories.

GRand Unified Bootloader (GRUB) An application used on most modern versions of the Linux operating system. It is a dynamically configurable program used to perform a sequence of events on a computer to load the main operating system. It receives control from the system BIOS, performs a sequence of events, and then transfers control to the operating systems kernel.

Graphical user interface (GUI) A design for the part of a program that interacts with the use and uses icons to represent programs features. GUIs typically work with "mouse-able" interfaces with pull-down menus, dialog boxes, check boxes, radio buttons, drop-down list boxes, scroll bars, scroll boxes, and the like.

Hard disk drive (HDD) A nonvolatile storage device that stores digitally encoded data on rapidly rotating platters with magnetic surfaces within a sealed unit.

Hypertext Transfer Protocol (HTTP) A TCP-based application-level protocol that is used to transfer hypertext requests and information between servers and browsers. HTTP uses TCP port 80.

Hypertext Transfer Protocol Secure (HTTPS) The secure version of HTTP that is encrypted using a cryptographic protocol (typically SSL or TLS) and uses TCP port 443.

Inode A data structure holding information about files in a UNIX filesystem. There is an inode for each file, and a file is uniquely identified by the filesystem on which it resides and its inode number on that system.

International Standards Organization (ISO) The world's largest developer and publisher of International Standards, it is a network of the national standards institutes of 161 countries, one member per country, with a Central Secretariat in Geneva, Switzerland, that coordinates the system.

Internet Message Access Protocol (IMAP) An e-mail protocol that allows users to receive using an IMAP e-mail client, where all e-mail, folders, and so on are stored on the server, rather than on the client's local computer.

Internet Software Consortium (ISC) A nonprofit, public benefit corporation dedicated to supporting the infrastructure of the universal connected self-organizing Internet – and the autonomy of its participants – by developing and maintaining core production quality software, protocols, and operations. It develops, maintains, distributes, and supports BIND, and ISC DHCP.

Java Virtual Machine (JVM) The runtime environment for executing Java code.

KDE Display Manager (KDM) The open-source graphical login interface from KDE, a nonprofit organization. Like GDM from the Gnome Project, it is the K Desktop Environment replacement for XDM, the default X display manager (from which it was originally developed). KDM allows users to pick their session type on a per-login basis.

Kernel The core operational code of an operating system. In Linux, it integrates the CPU architecture and supports the loading of modules and instructions to implement all operating system services (for example, process management, concurrency, and memory management).

Level 2 Transfer Protocol (L2TP) A tunneling protocol used to support virtual private networks (VPNs). It does not natively encrypt any information; it relies on a separate encryption protocol to secure data in transit within the tunnel.

Lightweight Directory Access Protocol (LDAP) A protocol that enables users to query and update information in a directory service.

Linux Loader (LILO) A bootloader for Linux.

lp A command that is used on many UNIX- and Linux-based computers to manage print jobs in printer queues.

Logical volume manager (LVM) A collection of programs that allows larger physical disks to be reassembled into "logical" disks that can be shrunk or expanded as data needs change.

Mail exchanger (MX) A DNS record type that is used to identify the authoritative MTA for a domain.

Mail transport agent (MTA) A program responsible for delivering e-mail messages. Upon receiving a message from a mail user agent or another MTA, it stores it temporarily locally and analyzes the recipients and either delivers it (local addressee) or forwards it to another MTA.

Mail user agent (MUA) A program that allows the user to compose and read e-mail messages. The MUA provides the interface between the user and the MTA.

man The command to format and display the online manual pages on a UNIX- or Linux-based system. It is short for "manual."

Mandatory Access Control (MAC) A type of access control where system privileges are specified by the system. They cannot be applied, modified, or removed – except perhaps by means of a privileged operation.

Media Access Control (MAC) A sublayer of the Data Link Layer specified in the seven-layer OSI model (Layer 2) that provides addressing (that is, a MAC address) and control mechanisms that enable multiple devices to communicate over a network.

Megabyte (MB) A measure of computer data storage where 1 MB is equal to 1,048,576 bytes.

MySQL An open-source relational database management system that is developed, distributed, and supported by Sun Microsystems, Inc.

Name Service Cache Daemon (NSCD) A daemon that provides a cache for the most common name service requests.

NESSUS A vulnerability scanner that was designed to automate the testing and discovery of known security problems.

Network File System (NFS) A protocol developed by Sun Microsystems, Inc. that allows a computer to mount a volume that resides on a remote computer and access files from across the network as if they were stored locally.

Network Information Service (NIS) A protocol for remote distribution of common configuration files developed by Sun Microsystems, Inc. It was originally called Yellow Pages (YP).

Network interface card (NIC) A built-in or peripheral adapter that is installed in a computer to provide a physical connection to a network.

Network mapper (NMAP) A security scanner that is used to scan for and discover computers and services on a network.

Network News Transfer Protocol (NNTP) A protocol that defines how news articles are passed around between computers.

Network Time Protocol (NTP) The TCP/IP protocol used to synchronize the clocks on computers across a network. NTP uses UDP on port 123.

New Technology File System (NTFS) A high-performance and self-healing file system proprietary to Windows XP Vista 2003 2000 NT & Windows 7, which supports file-level security, compression and auditing. It also supports large volumes and powerful storage solution such as RAID.

Operating system (OS) The software that provides the interface between hardware and user by managing and coordinating the activities and sharing of system resources.

Parent Process ID (PPID) The PID of the process that spawned (started) a new process.

Partition A logical section of a hard disk. Each partition normally has its own file system.

PHP (PHP: Hypertext Processor, formerly known as Personal Home Pages) A widely-used general-purpose scripting language that is especially suited for Web development and can be embedded into HTML.

Pluggable authentication module (PAM) A mechanism to integrate multiple low-level authentication schemes into a high-level application programming interface (API). It allows programs that rely on authentication to be written independently of the underlying authentication scheme.

Point to Point Protocol (PPP) A connection-oriented protocol for communication between computers using a serial interface, such as a computer using a modem, to connect to a server over a phone line.

Port (TCP/IP) A logical channel or channel endpoint in a communications system. Each application program has a unique port number associated with it. Port numbers distinguish between different logical channels on the same NIC.

Post Office Protocol (POP) An application-layer Internet standard protocol used by local e-mail clients to retrieve e-mail from a remote server over a TCP/IP connection.

Power performance computing (PPC) An RISC-based CPU architecture created by the 1991 Apple-IBM-Motorola alliance. PPC CPUs were used in IBM mid-range servers and Apple computers (until Apple's recent switch to Intel CPUs).

Process ID (PID) A unique number assigned when a new process (program) is started and used to reference that process.

Red Hat Package Manager (RPM) A software package management tool, developed by Red Hat, that is used on a variety of Linux distributions; the RPM file format is the chosen standard package format for the Linux Standard Base.

Redundant Array of Independent Disks (RAID) A form of technology available to Linux systems that uses your disk subsystem to provide enhanced read/write performance, protection against data lost due to disk failures, or both.

Regular expressions (regex) A recognized method for describing a search pattern.

ReiserFS A journaled filesystem that is supported in the Linux kernel.

Remote Authentication Dial-in User Services (RADIUS) An application-layer protocol that provides centralized Authentication, Authorization, and Accounting (AAA) management for computers to connect and use a network service.

Remote Desktop Protocol (RDP) A multichannel protocol that allows a user to connect to a computer running Microsoft Terminal Services (TS).

Route The path from a source device through a series of hosts, routers, bridges, gateways, and other devices that network traffic takes to arrive at a destination device on a different network.

Runlevel A specialized script that starts a different set of services, giving multiple different configurations in the same system.

Samba An open-source suite of programs that provides file and print services to SMB/CIFS clients and allows for interoperability between Linux/UNIX servers and Windows-based clients.

Sandbox A protected, limited environment where applications (for example, Java programs downloaded from the Internet) are allowed to "play" without risking damage to the rest of the system.

Secure copy (SCP) A command within the Secure Shell (SSH) suite that is used to securely transfer computer files between a local computer and a remote host or between two remote hosts.

Secure File Transfer Protocol (SFTP) A network protocol that uses SSH to securely transfer files while encrypting both commands and data.

Secure Shell (SSH) A network protocol that allows data to be exchanged using a secure channel between two computers.

Security Enhanced Linux (SELinux) A set of modifications that can be applied to UNIX- and Linux-based computers that provide the capability to manage security through the implementation of system security policies.

Server Message Block File System (SMBFS) A mountable Server Message Block filesystem for Linux that allows Windows- or Linux-based workstations access to directory/file shares on a network-based Linux server.

Shell (SH) A command line interpreter.

Single user mode A runlevel, usually runlevel 1, where logins are not allowed except by the root account. It is used either for system repairs or for moving filesystems around between partitions.

Simple Mail Transport Protocol (SMTP) A TCP/IP protocol that is used to transfer mail reliably and efficiently.

Simple Network Management Protocol (SNMP) An application-level protocol used to monitor and perform basic configuration of network devices.

Small Computer System Interface (SCSI) A set of standards for physically connecting and transferring data between computers and peripheral devices.

Snort An open-source network intrusion detection system for UNIX, Linux, and Windows.

Software package Software packaged in an archive format that is installed, managed, and removed using a package management system or stand-alone installation software.

Squid An open-source caching proxy server for the Web supporting HTTP, HTTPS, and FTP, among others.

Service Set Identifier (SSID) The name used to identify a wireless network.

Standard error (STDERR) An output stream from a script or a program that is the default destination for error messages and other diagnostic warnings and is directed to an output device, such as a terminal screen.

Standard input (STDIN) An input stream from a script or program that is the default source of data that will be used by the script or program. It is directed to an input device, such as a keyboard.

Standard output (STDOUT) An output stream from a script or a program that is the default destination of regular output for applications and is directed to an output device, such as a terminal screen.

Storage area network (SAN) A specialized network that provides access to high-performance and highly available storage subsystems using block storage protocols.

Swap The allocation of physical disk space to function as virtual memory when the amount of physical memory (RAM) is full. If the system needs more memory resources and the physical memory is full, inactive pages in memory are moved to the swap space. When the system swaps out pages of memory to the hard disk drive, the system's RAM is freed up to perform additional functions.

Time to Live (TTL) The "life expectancy," measured in time, iterations, or transmissions, of a unit of data before it is discarded.

Tomcat An open-source software implementation of the Java Servlet and Java Server Pages technologies produced by the Apache Software Foundation.

Tripwire An open-source intrusion detection system that monitors specified files and directories for unauthorized changes. There are commercial versions of Tripwire.

Two-factor authentication An authentication mechanism where two factors, such as a token and a password or a thumb print and a PIN, are combined to authenticate a user.

User ID (UID) A numeric identifier that represents a user, which is used to identify the user to various components within Linux.

Universal Serial Bus (USB) A serial bus standard to connect up to 127 peripheral devices to a host computer.

vi An advanced, highly configurable text editor that has been the de facto text editor on UNIX-based systems for decades. *vim* is an improved version on Linux systems that has a broader feature set.

Virtual File Allocation Table (VFAT) A Linux filesystem that is compatible with Windows 95 and Windows NT long filenames on the FAT filesystem.

Virtual Network Computer (VNC) Software that enables a user to remotely control a computer across a network connection.

Web proxy A server that acts as a go-between between a client and typically the Internet, often to perform filtering of the data. Squid is a good example of a Web proxy server.

Window manager A program or suite of software that controls the placement and appearance of windows within a windowing system in a GUI.

Wireshark An open-source network protocol analyzer for UNIX, Linux, and Windows.

X Window System (or simply, X) An open-source suite of software (including a network protocol) implements the X display protocol, provides windowing, and manages keyboard and mouse functions to provide a GUI for networked computers.

This page intentionally left blank

Index

T

U

V

W

X

Y

This page intentionally left blank